

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Utilisation d'imagerie thermique basse
résolution pour valider l'utilisation sécuritaire
d'une cuisinière électrique

Mémoire de maîtrise
Spécialité : génie électrique

Alexandre HUPPÉ

Jury : Philippe MABILLEAU (directeur)
Éric PLOURDE
Ruben GONZALEZ-RUBIO

RÉSUMÉ

L'utilisation d'appareils électriques, comme une cuisinière, par des personnes en perte d'autonomie représente souvent un risque important pour eux et leur environnement. Afin de réduire ce risque tout en permettant à ces personnes de poursuivre leurs activités, un projet de détection de danger par imagerie thermique a été mis sur pied.

Ce projet a permis de réaliser un prototype fonctionnel permettant d'évaluer le danger associé aux activités de cuisson à l'aide d'un traitement à base d'intelligence artificielle et d'en informer l'utilisateur.

Les résultats laissent présager que l'approche utilisée pourrait permettre de détecter des dangers de brûlure, ce qui représente un important gain sur les produits de sécurité existants.

Par exemple, l'approche utilisant la logique floue se montre très utile pour différencier les types de dangers alors que l'approche utilisant des réseaux de neurones offre une plus grande précision sur une détection binaire de danger ou sécurité.

À partir de ce prototype fonctionnel, un produit autonome, simple à installer et offrant une rétroaction appropriée à l'utilisateur sur le niveau de danger de son usage de la cuisinière pourrait être développé.

Mots-clés : sécurité, cuisinière, imagerie thermique basse résolution, sécurité des aînés, sécurité incendie, prévention des brûlures

TABLE DES MATIÈRES

1	INTRODUCTION	1
1.1	Mise en contexte	1
1.2	Positionnement du projet	2
2	ÉTAT DE L'ART	5
2.1	Contexte de recherche	5
2.2	Projets antérieurs	6
2.2.1	Prévention des dangers liés à l'utilisation d'une cuisinière : une validation grâce à un prototype	6
2.2.2	Approche à base de vérification formelle de modèle pour une utilisation sécuritaire de la cuisinière d'un habitat intelligent	8
2.2.3	Système autonome de sécurité lors de la préparation d'un repas pour des personnes cognitivement déficientes dans un habitat intelligent pour la santé.	8
2.3	Solutions existantes	9
2.3.1	Détecteurs de fumée	9
2.3.2	Autres appareils de sécurité	11
2.4	Types de danger	12
2.4.1	Incendies	12
2.4.2	Brulures	13
2.5	Intelligence artificielle	15
3	SOLUTIONS EXPLORÉES	19
3.1	Hypothèses de départ	20
3.2	Question de recherche	21
4	ALGORITHMIQUE	23
4.1	Capteur	23
4.2	Extraction de paramètres	25
4.3	Traitement	28
4.3.1	Logique floue	28
4.3.2	Réseau de neurones	36
4.4	Action	44
5	EXPÉRIMENTATIONS	47
5.1	Matériel utilisé	48
5.1.1	Montage utilisé	48
5.1.2	Système ordonné	52
5.2	Configuration logicielle	54
5.2.1	Acquisition d'images thermiques	57
5.2.2	Enregistrement des données	61

5.2.3	Scénarios d'expérimentation	63
5.2.4	Traitement des données	66
6	RÉSULTATS	71
6.1	Logique floue	71
6.2	Réseau de neurones	78
6.3	Adaptabilité	83
6.4	Tableaux synthèses	87
7	CONCLUSION	89
7.1	Limites et perspectives	91
A	Exemple de fichier de configuration	93
B	Description des paramètres	97
C	Configuration des GPIO du Raspberry Pi 2, modèle B	99
	LISTE DES RÉFÉRENCES	101

LISTE DES FIGURES

2.1	Plan du laboratoire DOMUS	6
2.2	Configuration du détecteur FCP-320/FCH-320	10
4.1	Schéma bloc du système	23
4.2	Caméra thermique basse résolution MLX90620	24
4.3	Diagramme fonctionnel du MLX90620	24
4.4	Pixels étant considérés voisins (en bleu) du pixel 6 (en rouge)	26
4.5	Exemple de blob (entouré de vert) lorsque le seuil de température est de 75°C	27
4.6	Diagramme de dispersion $\Delta_{blobs} - \Delta_{min}$	31
4.7	Exemple de situation sécuritaire - interface	32
4.8	Exemple de situation sécuritaire - logique floue	32
4.9	Exemple de situation avec rond exposé - interface	33
4.10	Exemple de situation avec rond exposé - logique floue	33
4.11	Exemple de situation avec début d'occlusion - interface	34
4.12	Exemple de situation avec début d'occlusion - logique floue	34
4.13	Exemple de situation avec occlusion détectée - interface	35
4.14	Exemple de situation avec occlusion détectée - logique floue	35
4.15	Configuration du réseau de neurones	37
4.16	Fonctionnement d'un neurone	37
5.1	Schéma du montage	48
5.2	Montage servant à l'enregistrement des données	49
5.3	Montage servant à l'enregistrement des données, vu de côté	50
5.4	Montage sans caméra	51
5.5	Raspberry Pi 2, Model B	52
5.6	Module caméra du Raspberry Pi	52
5.7	Schéma de l'architecture logicielle	54
5.8	Connexions matérielles entre le capteur MLX90620 de Melexis et le Raspberry Pi	57
5.9	Étapes d'utilisations du capteur MLX90620 de Melexis	60
5.10	Interface de visualisation	62
5.11	Interface permettant de rejouer un enregistrement préalable	62
5.12	Effet du biais sur la fonction sigmoïdale	68
6.1	Résultat de l'utilisation de logique floue sur « TestAI1 »	73
6.2	Résultat de l'utilisation de logique floue sur « TestAI2 »	74
6.3	Résultat de l'utilisation de logique floue sur « TestAI »	75
6.4	Agrandissement de la fin de la figure (des images 1200 à 1300) 6.3	76
6.5	Résultat de l'entraînement d'un réseau de neurones avec « TestAI1_subset »	79
6.6	Résultat de l'utilisation de réseaux de neurones sur « TestAI1 »	80
6.7	Résultat de l'utilisation de réseaux de neurones sur « TestAI2 »	81
6.8	Résultat de l'utilisation de réseaux de neurones sur « TestAI »	82

6.9	Résultat de l'utilisation des deux approches sur « cuisine2 »	84
6.10	Résultat de l'utilisation des deux approches sur « 2016-01-03_18-46-38 » .	86
C.1	Disposition des broches (« GPIO pinout »), Raspberry Pi 2, modèle B . .	100

LISTE DES TABLEAUX

2.1	Étapes de dégradation de l'huile de cuisson	13
5.1	Glossaire des variables utilisées dans les calculs du capteur	59
5.2	Liste des enregistrements utilisés pour le développement	63
5.3	Description des enregistrements utilisés	65
6.1	Écart moyen entre les valeurs évaluées manuellement, calculées et l'utilisa- tion de logique floue	88
6.2	Écart moyen entre les valeurs évaluées manuellement et l'utilisation d'un réseau de neurones	88

CHAPITRE 1

INTRODUCTION

1.1 Mise en contexte

Dans un contexte de population vieillissante, le problème de la perte d'autonomie devient de plus en plus répandu. En réponse à ce problème grandissant, il est primordial de développer des solutions pour adapter l'environnement autour de ces personnes afin de leur permettre de rester autonomes le plus longtemps possible tout en assurant leur sécurité.

En ce sens, ce projet vise à assurer la sécurité autour de la surface de cuisson d'une cuisinière électrique. Ce choix repose principalement sur le fait que la cuisine, plus particulièrement la cuisinière, a été démontrée comme la source principale de danger dans l'habitat [Ahrens, 2012, 2013; Ehrlich *et al.*, 2008, 2005; FEMA, 2004; Hall, 2006]. Pour ce faire, un prototype d'appareil de supervision de la surface de cuisson a été réalisé à l'aide d'imagerie thermique basse résolution.

Celui-ci a pour but de détecter, et si possible prévenir, les situations dangereuses pour l'utilisateur (de la surface de cuisson). Pour ce faire, nous avons dû cibler quels types de danger « surveiller » et déterminer l'action à poser suivant la détection de danger.

Ce qui est envisagé à ce sujet est, par exemple, de relier le prototype à une lampe dont l'intensité ou la couleur peut varier. Il serait alors possible de donner une rétroaction en temps réel à l'utilisateur.

De plus, il avait initialement été pensé qu'en cas de danger imminent, le prototype pourrait couper l'alimentation électrique. Pour ce faire, un module simple pourrait être branché entre la cuisinière et son alimentation électrique, ce qui limite les modifications du matériel existant. Bien que cette option ait l'avantage de limiter d'une façon drastique les risques, son caractère extrême ferait sans doute que la réaction naturelle de l'utilisateur serait probablement de trouver un moyen de contourner ce mécanisme (rebrancher la cuisinière directement à l'alimentation électrique, sans passer par le prototype), le rendant caduc.

Il est plutôt proposé de remplacer cette réaction drastique par la tenue d'un journal des événements. Celui-ci pourrait être consulté par les proches de l'utilisateur ou les intervenants en

vue de retirer l'accès à la cuisinière lorsque le niveau de danger sera trop grand (situations trop dangereuses et/ou trop fréquentes).

Même si le journal peut mener au retrait d'accès à la cuisinière, le but premier reste le même : permettre l'accès à la cuisinière aux personnes en perte d'autonomie, le plus longtemps possible, mais de façon sécuritaire. Ce journal permettrait également de limiter le caractère intrusif puisque l'utilisateur n'aurait pas à être constamment accompagné pour assurer sa supervision.

1.2 Positionnement du projet

Dans le cadre de ce projet, certains paramètres ont été fixés. Par exemple, le capteur qui a servi à la « surveillance » de la surface de cuisson devait être une caméra thermique basse résolution. De plus, ce capteur a été connecté à un petit système ordonné pour permettre l'acquisition et le traitement (dans le cadre des expérimentations, le traitement était effectué sur un autre ordinateur, mais il est tout à fait envisageable qu'ils puissent se faire sur le même appareil).

À l'origine, il avait été prévu que les dangers retenus pour ce projet seraient ceux de brûlure et d'incendies. En cours de route, le danger d'incendie a peu à peu été délaissé pour deux raisons :

1. Les travaux antérieurs sont plus dirigés vers les risques d'incendie.

En ce qui a trait aux brûlures, peu de solutions existent actuellement pour protéger les utilisateurs de cuisinière électrique. Pour ce qui est des incendies, il s'agit d'un sujet beaucoup plus couvert et pour lequel diverses solutions existent. Parmi ces solutions, le détecteur de fumée est la plus répandu[Ahrens, 2008; Istre et Mallonee, 2000], mais pas la plus efficace. Une nouvelle génération de ces appareils fait donc son apparition sur le marché. Par exemple, NEST offre un avertisseur de fumée et de CO₂[Nest Labs, 2014] qui intègre des avertissements avant l'alarme ainsi que diverses commodités telles que de parler à l'utilisateur pour indiquer, par exemple, dans quelle pièce se trouve le problème.

2. Il est plus difficile, dans une approche de prévention, de déterminer les limites à ne pas franchir.

J'entends par là quand avertir pour rester dans la prévention et non dans la réaction (si on prévient l'utilisateur trop tard nous ne sommes plus en prévention, mais plutôt en

réaction à un début d'incendie). De plus, quelles situations se trouvent à la frontière entre ce qui demeure sécuritaire et ce qui cause un incendie ? Cette question reste difficile à répondre et impossible à expérimenter dans les environnements qui étaient accessibles (pas d'installation spécialisée pour les risques d'incendie disponibles).

Cette solution se veut donc un appareil de sécurité dédié à la surface de cuisson de la cuisinière, ce qui en fera, nous l'espérons, une solution plus efficace tout en restant discrète, c'est-à-dire qu'elle n'entrave pas l'usage habituel de la cuisinière.

CHAPITRE 2

ÉTAT DE L'ART

2.1 Contexte de recherche

Ce projet a été réalisé dans le laboratoire DOMUS [[Laboratoire DOMUS, 2010](#)], un laboratoire de domotique de l'Université de Sherbrooke, dont les activités gravitent autour des personnes ayant divers troubles cognitifs. Le but étant de permettre aux personnes âgées ou avec des altérations des facultés intellectuelles de rester autonomes et à domicile plus longtemps sans dépendre exclusivement des aidants, naturels ou non. Par trouble cognitif on entend : les problèmes de mémoire, de jugement, de compréhension, de raisonnement, de concentration et/ou d'attention. Il est aussi possible d'y inclure certaines formes de démence, un trouble assez sévère pour perturber gravement la vie de tous les jours. Les pathologies concernées sont : l'Alzheimer et autre démence apparentée, la maladie de Parkinson et la maladie de Creutzfeldt-Jakob (vache folle).

Fait intéressant, dans le cas particulier de la maladie d'Alzheimer, l'espérance de vie peut atteindre les 20 ans après les premiers symptômes (bien que la moyenne soit de 4 à 6 ans) [[Alzheimer's Association, 2009](#)]. Ainsi, prolonger l'autonomie des personnes atteintes ne se veut pas un objectif seulement à court terme.

Le laboratoire DOMUS est un appartement intelligent (figure 2.1) intégrant une vaste panoplie de capteurs [[Bouchard, 2013](#)]. Toutes les informations acquises dans cet appartement peuvent donc servir à guider une personne en perte d'autonomie.

Cependant, malgré une forte présence de capteurs, il est primordial de réduire l'intrusivité au minimum. En effet, si certains groupes de recherche ont opté pour utiliser du vidéo pour la détection d'activités [[Geoff West et Greenhill, 2005](#)], au DOMUS, la seule caméra en place a pour but d'enregistrer les expérimentations. Ses données ne contribuent donc pas aux algorithmes qui y sont développés. Notons que de ne pas utiliser de vidéo et de réduire au minimum l'usage de l'audio n'empêche pas la télésurveillance, mais la rend moins intrusive.

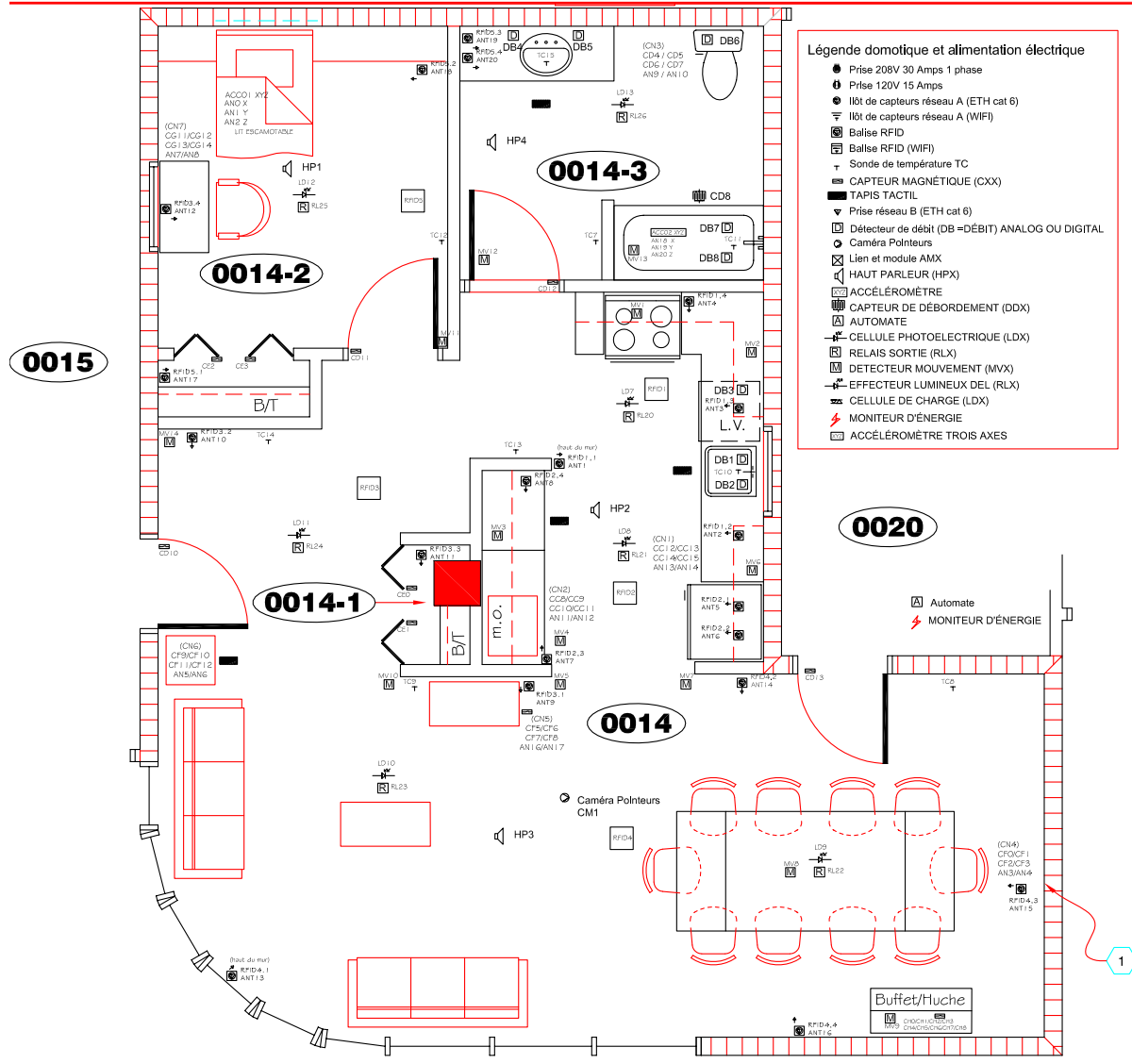


Figure 2.1 Plan du laboratoire DOMUS

2.2 Projets antérieurs

2.2.1 Prévention des dangers liés à l'utilisation d'une cuisinière : une validation grâce à un prototype

Le projet actuel s'inscrit directement à la suite de la maîtrise de Thomas Tessier [Tessier, 2012] qui explorait les différentes possibilités pour assurer la sécurité autour de la cuisinière électrique. Pour ce faire, il a réalisé une étude sur les solutions de surveillance disponibles sur le marché afin de déterminer quelles informations permettent de superviser

adéquatement une surface de cuisson pour prévenir le plus de dangers possible. Parmi ces dangers, notons les risques de brûlures, d'incendies et d'intoxications.

À la suite de son exploration, une preuve de concept avait été réalisée à l'aide de différents capteurs : capteur de présence, d'humidité, de température et de gaz (plusieurs types). Ce prototype, bien que convaincant, n'atteignait que partiellement les objectifs initiaux :

1. Détermination des informations à surveiller
2. Intégration dans l'environnement
3. Prise de décision sur la dangerosité de la situation en temps réel

L'objectif qui a été le moins bien atteint est sans aucun doute celui de l'intégration dans l'environnement.

En effet, les différents capteurs étant répartis autour et au-dessus de la surface de travail devenaient encombrants, difficiles à dissimuler, voire même gênants, pour l'utilisateur. Par exemple, il n'était pas possible d'utiliser la hotte ou la lampe de celle-ci lors de l'utilisation du prototype d'appareil de surveillance. Il n'était également pas possible d'utiliser plus d'un élément chauffant, le prototype n'étant pas en mesure de surveiller la surface de travail au complet.

Un autre aspect limitatif était lié à l'initialisation des capteurs. En effet, avant d'être complètement fonctionnels, certains capteurs (de gaz) pouvaient demander jusqu'à une semaine pour se stabiliser. De plus, les mesures thermiques effectuées à l'aide d'un capteur infrarouge étaient biaisées par l'environnement. Ce biais s'explique facilement considérant qu'un seul point de mesure couvrant une surface aussi grande qu'un élément chauffant ne mesure pas seulement la température de l'élément, mais aussi la température de l'air et des ustensiles environnants (à l'intérieur de son « point de mesure »).

Il est donc proposé de faire suite à ce projet en limitant le nombre et le type de capteurs afin d'atteindre l'objectif initial d'une bonne intégration dans l'environnement de travail de la cuisinière. Pour ce faire, nous prévoyons utiliser une caméra thermique basse résolution et un petit système ordonné.

2.2.2 Approche à base de vérification formelle de modèle pour une utilisation sécuritaire de la cuisinière d'un habitat intelligent

Le projet de [Tessier](#) n'a pas été le seul projet visant à assurer la sécurité autour de la cuisinière au DOMUS. En effet, [de Champs](#) a effectué en parallèle un projet sur l'aspect logiciel de la sécurité de la surface de cuisson.

Dans ce projet, [de Champs](#) nous propose un recensement des risques, principalement pour une personne âgée ou avec des déficits cognitifs, qui découlent des activités de la vie quotidienne. Parmi les risques évoqués, nous retrouvons : les brûlures, les incendies, les intoxications, les chutes, la médication, l'alimentation ainsi que l'errance et la désorientation.

Bien entendu, la majorité de ces risques ne seront pas retenus dans le cadre du projet actuel puisqu'ils ne sont pas directement en lien avec l'utilisation de la cuisinière ou ne peuvent être détectés à l'aide d'une caméra thermique basse résolution positionnée dans l'environnement immédiat de la cuisinière.

[de Champs](#) évoque aussi d'autres faits intéressants. Par exemple, comme la population mondiale est vieillissante, des tendances démontrent [[Bharucha et al., 2009](#)] que le nombre de personnes atteintes de déficits cognitifs augmentera considérablement dans les prochaines années. Il s'agit d'un fait intéressant puisqu'il est démontré [[Mallonee et al., 1996](#)] que certains groupes d'âge sont plus à risque de déclencher un incendie. Notons par exemple les enfants, les personnes âgées ainsi que celles nécessitant d'importantes médications. De plus, ces mêmes personnes sont les plus susceptibles de ne pas être en mesure d'entretenir les solutions de sécurité existantes telles que le détecteur de fumée [[de Champs, 2012](#)].

2.2.3 Système autonome de sécurité lors de la préparation d'un repas pour des personnes cognitivement déficientes dans un habitat intelligent pour la santé.

Plus récemment, [Kuijpers et al.](#) a terminé un projet de plus grande envergure : utiliser un système multi agent adaptatif pour assister une personne avec déficience lors de la préparation des repas. Pour ce faire, divers agents intelligents travaillent ensemble pour tenir compte de la situation et intervenir de façon appropriée.

Par exemple, selon le nombre de personnes présentes et leur profil (aidant ou patient avec divers degré de déficience), l'accès aux divers appareils peut être restreint.

Aussi, si une situation potentiellement dangereuse est détectée, le système demande à l'utilisateur de corriger la situation. Si l'utilisateur ne réagit pas dans le délai prévu, le système peut alors bloquer l'accès à l'appareil en cause pour limiter les risques pour l'utilisateur et son environnement.

Ce système est par contre beaucoup plus complexe que celui que nous proposons. Par exemple, le nombre d'appareils et d'agents inclus dans le système peut varier : des agents peuvent s'ajouter ou se retirer et c'est au système de s'adapter de façon autonome.

Une avenue qui pourrait être intéressante à explorer serait d'incorporer le prototype que nous proposons en tant qu'agent participant au système de [Kuijpers et al.](#).

2.3 Solutions existantes

Afin de bien situer ce projet comme étant complémentaire aux solutions existantes, il est proposé ici de considérer ce qui est actuellement disponible commercialement.

2.3.1 Détecteurs de fumée

La solution la plus répandue et de loin la plus connue est sans aucun doute le détecteur de fumée. Bien que généralement efficace pour détecter l'effet d'un feu, la fumée, il comporte des inconvénients. Par exemple, il est sujet aux fausses alarmes : qui n'a pas déjà déclenché son détecteur de fumée en utilisant un appareil de cuisine (four, grille-pain ou autre) alors que tout était pourtant sous contrôle et qu'aucun feu ne se déclarait ? Ainsi, certaines personnes choisissent de rendre inopérant leur détecteur de fumée pour le faire taire alors que dans d'autres cas la pile n'est pas remplacée. On remarque donc que malgré qu'ils soient présents dans presque toutes les habitations, les détecteurs de fumée ne sont pas tous pour autant fonctionnels [[Ahrens, 2008](#)].

De plus, ces appareils sont fondamentalement conçus pour nous permettre de réagir rapidement lorsqu'un incendie se déclare et non pas d'éviter celui-ci. On cherche donc à réagir plutôt que prévenir.

Note : Selon si on considère une réaction rapide pour éteindre le début d'incendie ou si on considère la prévention de perte de vie ou blessure par l'évacuation, un détecteur de fumée peut être perçu comme un moyen de prévention ou de réaction rapide. Le problème majeur avec ce type d'appareil réside dans la fréquence des faux positifs. Heureusement, la tendance actuelle est à minimiser ces fausses alarmes en utilisant un 2e niveau d'avertissement ou la possibilité

de faire taire l'alarme. Par exemple, le Nest Protect[Nest Labs, 2014] propose un avertissement verbal avant de déclencher une véritable alarme si aucune réponse n'est fournie par l'utilisateur dans un délai raisonnable. Dans le cadre de ce projet, il nous faudra aussi minimiser les faux positifs, probablement aussi avec plusieurs niveaux d'avertissements et alarmes.

Bien entendu, les détecteurs de fumées se sont améliorés au fil des années et ont été le sujet de plusieurs recherches. Par exemple, le fait de combiner plusieurs types d'informations et de les analyser avec des circuits plus complexes a donné lieu à des produits de détection intéressants. Par exemple, Bosch[Bosch Sicherheitssysteme GmbH, 2014] propose actuellement 2 modèles (FCP-320/FCH-320[Bosch Sicherheitssysteme GmbH, 2017a] (figure 2.2) et FCP-500[Bosch Sicherheitssysteme GmbH, 2017b]) de détecteurs d'incendies (notons le nom détecteur d'incendies plutôt que de fumée) qui combinent des capteurs optiques, thermiques et chimiques (gaz). Ces capteurs sont donc bien plus efficaces dans le sens où ils causent beaucoup moins de fausses alarmes.

- 1 Chambre de détermination de l'opacité des fumées avec capteur optique
- 2 Capteur thermique
- 3 Capteur chimique (recouvert sur l'illustration)
- 4 Témoin lumineux
- 5 Carte de circuit imprimé avec circuit électronique d'analyse
- 6 MS 400/MS 400 B - Socle

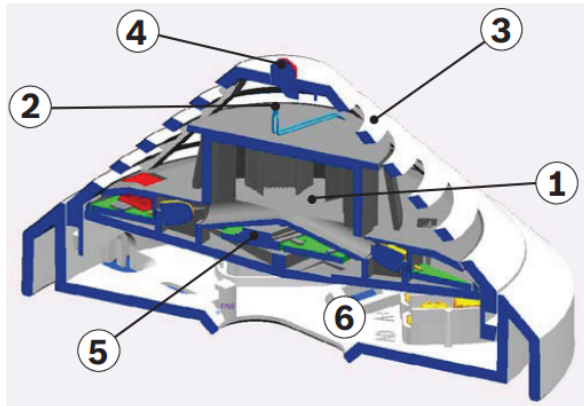


Figure 2.2 Configuration du détecteur FCP-320/FCH-320¹

On remarque donc qu'une meilleure connaissance de l'environnement permet généralement une meilleure interprétation des signes [Chen *et al.*, 2003; Gottuk *et al.*, 2002; Pfister, 1997; Rose-Pehrsson *et al.*, 2000]. Ainsi, nous espérons pouvoir obtenir une détection de qualité en ayant 64 mesures de température pour la surface de cuisson d'une cuisinière électrique. De plus, notre approche vise à prévenir plutôt que réagir.

¹Figure 2.1 du guide d'opération : http://resource.boschsecurity.com/documents/Operation_Guide_frFR_1267955979.pdf

2.3.2 Autres appareils de sécurité

Afin de ne pas trop s'écarter du sujet principal, seules les solutions reliées à la sécurité autour de la cuisinière seront abordées.

En dehors des indicateurs de surface chaude sur les cuisinières, il existe quelques autres solutions qui ciblent la prévention. Par exemple, *StoveGuard*, une solution qui supervise la température autour de la cuisinière, permet de couper son alimentation lorsqu'il y a risque d'incendie [Crowley, 2004]. Ce produit aurait probablement été raffiné par la suite puisque dans le mémoire de mon prédécesseur [Tessier, 2012] celui-ci mentionne le même produit, *StoveGuard*[Protection E.M. Inc., 2012], qui détecte la présence d'une personne à proximité de la cuisinière. Lorsque personne n'est assez près, l'alimentation de cette cuisinière est temporairement interrompue jusqu'à ce qu'une présence soit de nouveau détectée.

Cette solution est intéressante, mais comporte un inconvénient majeur : l'absence d'une personne n'est pas nécessairement un indicateur de danger et au contraire, une personne présente ne garantit pas non plus la sécurité. Cette affirmation est d'autant plus vraie lorsqu'il s'agit d'une personne atteinte de déficit cognitif[de Champs, 2012]. Cependant, la version mentionnée par Crowley [Crowley, 2004] semble plus proche du projet actuel en ce sens qu'on y surveille la température et prend action si le danger devient trop élevé. Par contre, il n'est question que de prévention d'incendie alors que nous prévoyons que notre solution pourra s'intégrer dans un système d'assistance cognitive, par exemple un habitat intelligent.

D'autres solutions potentielles existent, par exemple : *HomeSenser*[HomeSense Enterprises, 2006] ainsi que *SafeCook*[Protection E.M. Inc., 2014] sont deux solutions qui permettent de limiter le temps d'utilisation sans supervision de la cuisinière avant de couper son alimentation électrique.

Il existe également d'autres solutions plus évoluées telles que le *Safe-T-Element*[Bonner, 2007; de Champs, 2012; Farmer, 2012; Paszczuk, 2009; Pioneering Technology Corp., 2011; Tessier, 2012]. Il s'agit d'un appareil simple d'utilisation qui supervise la température d'un élément de surface de cuisson en vue de couper son alimentation en cas de dépassement d'un seuil. Comme il est reconnu que la température d'ignition de l'aliment le plus dangereux, l'huile, se situe autour de 350°C, il est donc possible, en limitant l'apport en chaleur, de réduire fortement les risques d'incendie. De plus, comme il s'agit d'un appareil que l'on installe sur la surface de cuisson, il se fait discret et n'entrave donc pas les activités culinaires.

Cette solution est très intéressante puisque tout comme pour le projet actuel, il s'agit de surveillance de température de la surface de cuisson en vue d'agir si nécessaire pour prévenir le danger. Par contre, comme il n'est question que de prévention des incendies ; la protection contre les brûlures ou autres situations dangereuses n'est pas abordée. L'avantage de cette solution s'avère être la précision de mesure de la température puisque contrairement à une mesure à distance avec de l'imagerie thermique, il s'agit d'une mesure à la source.

On note aussi que les mêmes auteurs mentionnent une autre solution : *StoveTop FireStop* [Bonner, 2007; de Champs, 2012; Farmer, 2012; Paszczuk, 2009; StoveTop FireStop, 2014; Tessier, 2012]. Cette solution est un peu différente au sens où il n'y a pas vraiment de supervision, mais une réaction qui est produite. On ne parle donc pas de prévention. Le principe est simple, lorsqu'un feu se déclenche sur la cuisinière, les flammes seront généralement assez hautes pour atteindre la hotte. Ainsi, en y installent un boîtier sensible à la chaleur et rempli d'une poudre sèche, on peut éteindre un début d'incendie avant qu'il ne prenne des proportions difficiles à contrôler.

Cette solution est intéressante, mais agit en réaction plutôt qu'en prévention. Cependant, il est intéressant de noter que tout comme un gicleur, aucune intervention humaine ne serait nécessaire pour maîtriser un début d'incendie.

2.4 Types de danger

Voici un bref aperçu de ce que la revue de littérature a permis de dégager sur les dangers qui seront considérés dans le cadre de ce projet. Notons que de Champs a effectué dans son mémoire un travail beaucoup plus large sur les différents risques que l'on peut rencontrer lors des activités de la vie quotidienne. Nous ne reprendrons donc pas ici un inventaire aussi complet².

2.4.1 Incendies

Comme nous l'avons vu à la section 2.3, la majorité des solutions existantes ciblent ce danger. Par contre, elles sont majoritairement en réaction plutôt qu'en prévention.

²Voir le tableau 1.1 du mémoire de Thibault de Champs pour une synthèse des risques à domicile [de Champs, 2012]

La cause principale d'incendie reliée à l'utilisation d'une cuisinière est l'huile [Wijayasinghe et Makey, 1997]. En effet, celle-ci, lorsqu'oubliée sur le feu est particulièrement dangereuse parce qu'elle se transforme chimiquement.

En augmentant, la température fait atteindre diverses étapes à l'huile, tel que décrit au tableau 2.1. On y constate que selon le type d'huile, elle atteindra plus ou moins rapidement différentes étapes critiques. Par exemple, le point de fumée (« Smoke Point ») est le moment où commencent à se volatiliser les produits de dégradation de l'huile. À une température généralement plus élevée, nous atteignons le point d'éclair (« Flash Point »), moment où une étincelle peut enflammer les vapeurs. Heureusement, si la source d'ignition est retirée, les vapeurs cessent de brûler. Après ce point, le point de feu (« Fire Point »), est le moment où même si l'on retire la source des flammes, celles-ci se poursuivent. Finalement, le point d'auto-ignition est quand la température est telle que les vapeurs peuvent s'enflammer sans aucune source externe [Tessier, 2012].

Il est intéressant de noter ici que le capteur que nous prévoyons utiliser ne peut mesurer de température supérieure à 300°C. Il est supposé que toute température supérieure à ce seuil serait, de toute façon, un signe de danger. Ainsi, nous croyons que cette limite n'empêchera pas l'obtention de bons résultats.

Étape	Température
« Smoke Point »	220-240°C
« Flash Point »	220-321°C
« Fire Point »	non disponible
« Autoignition Temperature »	315-445°C

Tableau 2.1 Étapes de dégradation de l'huile de cuisson[Wijayasinghe et Makey, 1997]

2.4.2 Brulures

Tout d'abord, qu'est-ce qu'une brûlure ? On peut définir celle-ci comme étant une destruction des cellules de la peau, que ce soit suite à une exposition à des températures trop élevées, des flammes, des radiations ou même des produits chimiques. Selon la gravité d'une brûlure, on dira de celle-ci qu'elle est de 1er, 2e ou 3e degré.[Tessier, 2012]

Pour bien comprendre ce que représentent ces degrés, prenons quelques instants pour comprendre comment la peau et la chaleur se comportent.

La peau humaine est composée de couches : l'épiderme et le derme. Ce sont ces 2 couches qui protègent les autres éléments de notre corps du monde extérieur.

La température, quant à elle, est une forme d'énergie transmise qui cherche à s'équilibrer entre les éléments qui entrent en contact. Que ce soit un contact direct ou indirect, il peut y avoir transfert de chaleur par conduction, convection ou radiation. Nous n'entrerons pas dans les détails des différentes formules régissant ces trois types de transfert, mais il est important de retenir qu'un contact direct va provoquer un transfert par conduction, par exemple lorsque l'on touche un ustensile de cuisson chaud, nous absorbons sa chaleur.

Ce transfert de chaleur étant une recherche d'équilibre (le corps le plus chaud réchauffe l'autre), plus il y a un grand écart de température entre les corps qui participent à l'échange, plus il y aura un grand transfert. De plus, tous les corps n'échangent pas la chaleur à la même vitesse. Il existe en effet une valeur pour chaque corps, nommé conductivité thermique, qui régit à quelle vitesse le transfert a lieu[[Henriques et Moritz, 1947](#)].

Il existe aussi une autre valeur importante, la chaleur massique. Pour chaque corps, cette valeur permet de déterminer la quantité de chaleur nécessaire pour augmenter la température d'un gramme de ce corps d'un degré.

Sachant cela, que représente une brûlure au 1er degré ? Toute brûlure qui détruit en partie l'épiderme. On parle ensuite de 2e degré si l'épiderme est complètement détruit et de 3e degré si le derme est complètement détruit[[Moritz, 1947](#)].

Maintenant, comment se produit cette destruction ? Nous savons que la destruction de cellules, les brûlures, peut avoir plusieurs causes. Dans notre cas, concentrons-nous sur celles causées par la chaleur.

Une cellule est susceptible d'être détruite lorsque sa température atteint les 45°C[[Wartzek et al., 2011](#)]. Comment cela se produit-il ? Simplement en ayant un contact de durée suffisante avec une source de chaleur pour que la peau gagne suffisamment de chaleur.

Par exemple, si notre main entre en contact avec une casserole chaude et reste en contact assez longtemps, la chaleur de la casserole sera transmise à la peau de notre main et en fera augmenter la température. Comme l'épiderme est la couche la plus à l'extérieur, celle-ci se réchauffera en premier. Une fois sa température suffisamment augmentée, elle transmettra à son tour de la chaleur au derme, qui lui aussi se réchauffera.

Ainsi, selon la durée d'exposition, la conductivité thermique et la chaleur massique de la peau, une brûlure peut être plus ou moins grave [[Henriques et Moritz, 1947](#); [Moritz, 1947](#); [Moritz et Henriques, 1947](#); [Tessier, 2012](#); [Wartzek et al., 2011](#)]. Retenons simplement

que plus la source est chaude, plus le danger est grand (le transfert de température sera plus rapide). Aussi, plus l'exposition à la source sera longue, plus les dommages seront importants.

Pour ce qui est de notre projet, les causes de brûlures les plus probables seront les ustensiles (chaudrons et casseroles par exemple) de cuisine chauds et les plaques de cuisson sans ustensiles. Ces éléments seront considérés dangereux à partir de 70°C puisqu'à partir de cette température, un contact de 1 à 2 secondes peut brûler jusqu'au 3e degré.

Nous savons que s'il s'agit d'un contact direct il y a moins de risques puisque les réflexes nous feront nous éloigner de la source de chaleur rapidement. Cependant, pour une écla-boussure, le temps nécessaire pour retirer cette source collée à la peau peut facilement laisser quelques secondes d'exposition et donc créer une blessure plus grave.

Fait intéressant pour nous, comme nous savons que la chaleur se propage toujours du corps le plus chaud vers un corps plus froid, nous pouvons déduire que la température d'un ustensile ne dépassera jamais la température de la surface de cuisson. Nous pouvons également déduire que la température du contenu de l'ustensile ne dépassera pas la température du contenant.

Ainsi, en mesurant la température de la plaque de cuisson (aux endroits non recouverts) et des ustensiles de cuisson, nous pouvons estimer efficacement la température du contenu, ce qui constitue une information suffisante pour interpréter le niveau de danger de la situation [Tessier, 2012].

2.5 Intelligence artificielle

L'intelligence artificielle est un domaine relativement nouveau (dans le domaine des sciences). Le terme serait apparu en 1956 et les premiers développements publiés auraient commencé vers la fin de la Seconde Guerre mondiale [Negnevitsky, 2005; Russell *et al.*, 2010].

Parmi les définitions de l'intelligence artificielle, on retrouve plusieurs concepts centrés autour de l'humain et de la rationalité. En effet, pour certains, l'intelligence est définie par un raisonnement et des actions qui sont comparables à ceux de l'être humain alors que pour d'autres, tout repose sur la capacité d'utiliser un modèle pour en venir aux meilleurs conclusions et actes possibles en fonction des informations disponibles [Russell *et al.*, 2010].

Dans le cadre de ce projet, la seconde approche sera privilégiée puisque nous cherchons à identifier le plus efficacement possible les situations dangereuses sans nécessairement nous limiter à l'analyse qu'en ferait un humain à partir des mêmes données.

Ceci étant dit, comment peut-on reconnaître une situation dangereuse ? Il a été démontré que pour faire une classification (danger par opposition à sécuritaire), il faut généralement avoir une bonne connaissance de la situation [Pfister, 1997; Tessier, 2012]. Par exemple, dans notre cas, il nous faudrait une grande quantité de données sur toutes les situations possibles, sécuritaires ou non, pour l'utilisation de la surface de cuisson d'une cuisinière électrique. Il nous faudra également avoir accès à l'information utile pour caractériser la situation actuelle. C'est ici qu'interviennent les capteurs (une caméra thermique basse résolution) que nous utilisons. Ce capteur nous procure donc de l'information sur la situation en cours que l'on peut ensuite utiliser pour estimer le niveau de danger.

En plus de l'information sur le moment présent, il est également possible de considérer l'évolution de l'information fournie par nos capteurs dans le temps. Ainsi, si notre prototype remarque une augmentation de température, il est possible d'en tirer une conclusion différente que s'il s'agit d'une température constante.

Comme l'information sur les situations dangereuses est difficile à acquérir et que le DOMUS n'est pas équipé pour effectuer des tests dans de telles situations, nous sommes dans l'impossibilité d'avoir une connaissance complète des scénarios possibles. Nous devons donc nous tourner vers d'autres solutions.

Heureusement, plusieurs auteurs ont déjà exploré ce type de problématique [Dreiseitl *et al.*, 2010; Fischer, 2004; Yin *et al.*, 2008]. Selon leurs recherches, dans divers domaines, il est possible d'effectuer une détection d'anomalies plutôt qu'une classification pour les problèmes où les situations qualifiées d'anormales sont peu fréquentes et/ou peu d'informations sont disponibles sur ces anomalies.

Dans le cadre de ce projet, nous pouvons considérer, par leur faible occurrence, que les situations dangereuses ne seront pas fréquentes dans l'usage de la cuisinière. Du moins, l'apprentissage ou le paramétrage de nos algorithmes se fera dans un environnement ne permettant pas de créer des situations dangereuses. Il nous sera donc possible d'accumuler des données d'utilisation sécuritaire et de déterminer que toute situation qui s'écarte trop du contexte d'apprentissage peut être considérée comme dangereuse.

Notons que selon l'utilisateur et ses habitudes de cuisine, les données recueillies par notre capteur ne seront pas toujours les mêmes. De cette réalité découle que notre apprentissage doit se poursuivre après une phase initiale (avant d'être disponible à l'utilisateur) ou se faire

entièrement avec l'utilisateur. De plus, comme il se peut qu'une personne en visite (ou autre situation similaire) utilise la cuisinière, de nouvelles habitudes pourraient être exposées à notre système, qui pourrait alors considérer la situation comme anormale.

Nous devons donc être en mesure de considérer les situations anormales comme potentiellement dangereuses, idéalement avec un niveau de certitude (niveau de confiance). De cette façon, toute situation anormale ne sera pas retenue comme dangereuse, mais uniquement celles qui s'éloignent trop (avec un seuil explicite ou non, selon le type d'algorithme utilisé) d'une situation normale.

Ainsi, pour réaliser une telle détection d'anomalies, il est possible qu'en étudiant les données nous soyons en mesure de trouver une frontière entre les situations sécuritaires et dangereuses. Dans une telle éventualité, cette frontière permet de trouver une limite pour un ou des critères et donc de créer des règles simples pour classer les situations.

Par contre, dans l'éventualité où une telle frontière ne peut être facilement extraite des données, nous pourrions utiliser des algorithmes de plus haut niveau tel que l'algorithme des k-Moyenne [Tessier, 2012].

Fischer énonce également diverses techniques qui sont intéressantes à explorer telles que l'utilisation de réseaux de neurones, une technique qui cherche à utiliser un fonctionnement similaire au cerveau humain. Il insiste cependant sur le fait qu'un bon prétraitement des données est souvent nécessaire pour limiter la taille d'un réseau performant. Dans un cas où un tel prétraitement ne peut être effectué, une grande quantité de données d'apprentissage doit être fournie puisqu'un réseau contenant plus de neurones doit nécessairement avoir plus de données d'entraînement pour être performant. Il est intéressant de noter qu'un plus grand réseau de neurones pourra compenser l'absence de prétraitement en extrayant lui-même les informations utiles de ses entrées.

Fischer propose également l'utilisation de la logique floue lorsqu'il est possible d'exploiter des connaissances sur le domaine pour générer des règles. Cette branche d'intelligence artificielle a l'avantage d'être plus facile à approcher par l'humain, mais n'est pas toujours indiquée puisque ces connaissances doivent être issues de l'expertise d'un humain. Il est donc déconseillé de générer automatiquement, par un algorithme quelconque, ces règles floues [Fischer, 2004].

CHAPITRE 3

SOLUTIONS EXPLORÉES

Tel que mentionné précédemment, il existe peu de solutions commercialement disponibles qui permettent d’assurer la sécurité autour de la cuisinière dans une optique de prévention. Ce nombre diminue encore plus lorsque le but est de permettre à des personnes avec des troubles cognitifs ou des pertes d’autonomie de continuer d’utiliser, sécuritairement, leur cuisinière.

Ce projet cherche donc à réaliser un prototype d’appareil de supervision intelligent qui facilite la prévention des divers dangers entourant l’utilisation de la surface de cuisson d’une cuisinière électrique par une rétroaction immédiate ainsi qu’avec l’aide d’un journal d’évènement pouvant être évalué par les aidants de l’utilisateur. De plus, nous ne souhaitons pas entraver l’utilisation normale de la cuisinière ni encombrer son environnement immédiat.

Pour ce faire, nous souhaitons utiliser le moins de capteurs et d’instruments possibles. Nous faciliterons ainsi l’installation, l’utilisation et conserverons un caractère non intrusif. Bref, nous avons en tête de développer une solution permettant de répondre à un besoin qui n’était pas traité adéquatement par les solutions disponibles.

En ce qui a trait au capteur utilisé, il avait été proposé en début de projet de valider si un capteur thermique basse résolution pouvait permettre d’identifier des situations dangereuses et si oui, quelles étaient ses limites. Nous avons cependant rapidement fait le constat qu’il n’existe pas beaucoup de ce type de caméras. C’est pourquoi nous avons choisi le MLX90620[Melexis, 2014] de Melexis (plus de détails à la section 4.1).

De plus, comme nous cherchions à faciliter l’intégration de notre solution dans un système plus large (habitat intelligent par exemple), nous cherchions, lors de la conception de notre prototype, à faciliter la communication potentielle avec le monde extérieur. Cependant, ce type d’intégration n’étant pas le but principal, elle n’a pas été implémentée dans le cadre de ce projet, mais la conception a tout de même été réalisée pour faciliter une éventuelle intégration (plus de détails à la section 5.2).

3.1 Hypothèses de départ

Le projet de Thomas Tessier [Tessier, 2012] nous permettait d’anticiper de bons résultats. En effet, bien que ses différents capteurs (gaz, température, distance, etc.) ne permettaient pas de couvrir toute la surface de cuisson, il a réussi à obtenir des résultats très intéressants. Nous anticipions donc être en mesure de surveiller adéquatement la surface de cuisson en entier considérant que nous avons beaucoup plus de points de mesure de température (64 pixels). Cependant, nous étions conscients que nous n’aurions pas les autres types de capteurs dont le projet précédent bénéficiait. Toutefois, nous estimions être en mesure, uniquement avec les caméras thermiques, d’obtenir l’information nécessaire (ex. : la température, quelles sections de la plaque de cuisson sont utilisées et possiblement de capter les mouvements de l’usager). De plus, nous estimions qu’il était possible de réduire considérablement le biais en augmentant le nombre de points de mesure. Rappelons à ce sujet que le projet de Thomas Tessier [Tessier, 2012] ne disposait que de capteurs de température ne retournant qu’une seule valeur. Ainsi, les mesures prises tenaient compte non seulement de la température de l’objet visé, mais aussi la température ambiante autour de l’objet. La lecture était donc une moyenne de ce que le capteur pouvait « voir » et non pas la température de l’objet.

Il était envisagé que d’utiliser une caméra thermique, même basse résolution, nous permettrait d’augmenter notre précision et de dégager des patrons (« patterns ») de température sur la surface de cuisson.

Il restait cependant à confirmer que ces patrons, couplés avec des algorithmes d’intelligence artificielle, étaient suffisants pour différencier les situations dangereuses des situations normales.

À ce sujet, voici une liste non exhaustive des situations dangereuses que nous cherchions à identifier :

- La température d’une partie (ou totalité) de la surface de cuisson atteint un seuil critique.
- Une plaque de cuisson qui n’est pas recouverte par un ustensile de cuisson est activée (risque de brûlure).
- Une partie du corps de l’usager est trop longtemps immobile près d’une source de brûlure potentielle (ustensile de cuisson par exemple).

- Une élévation de température localisée trop rapide (pourrait indiquer un départ de feu : linge oublié en contact avec une source de chaleur par exemple).

3.2 Question de recherche

Dans cette optique, nous posons la question suivante : *Est-ce qu'un patron de température peut permettre d'identifier des situations dangereuses lors de l'utilisation de la surface de cuisson d'une cuisinière électrique ?*

De cette question en découlent certaines autres qui pourraient être adressées dans le futur, mais qui ont quand même guidé les expérimentations :

- Quelles sont les limites de température acceptables en toute sécurité ?
- Ces limites peuvent-elles être caractérisées dans certains patrons (spatio-temporels) ?
- Ces températures et ces patrons sont-ils détectables avec les capteurs ciblés ?
- Est-il utile de superviser l'espace environnant immédiat de la surface de cuisson (comptoir par exemple) pour détecter ces patrons ?
- Est-il possible de détecter la présence d'une main (ou de toutes autres parties du corps) au-dessus (ou à proximité) de la surface de cuisson et de zones de température élevée ?

CHAPITRE 4

ALGORITHMIQUE

Comme il en était question à la section 3.2, nous cherchons à valider si l'utilisation d'une caméra thermique à basse résolution peut nous permettre d'identifier des situations dangereuses liées à l'utilisation d'une cuisinière électrique. L'une des pistes explorées est l'utilisation de patron de température, donc de « formes » créées par les différentes températures dans l'image.

Pour tester notre question de recherche, le système de la figure 4.1 a été utilisé. Il comporte une phase d'acquisition des données du capteur, une transformation des données brutes en paramètres plus significatifs (par exemple un maximum ou une moyenne) pour ensuite laisser place au traitement développé pour déterminer le niveau de danger d'une situation, le tout menant à une action de la part du système.

Le présent chapitre s'intéressera spécifiquement au capteur ainsi qu'aux approches théoriques utilisées pour le traitement des données acquises, notamment la logique floue et l'utilisation d'un réseau de neurones. L'intégration du capteur et de la chaîne de traitement dans un montage physique sera abordée au chapitre 5.

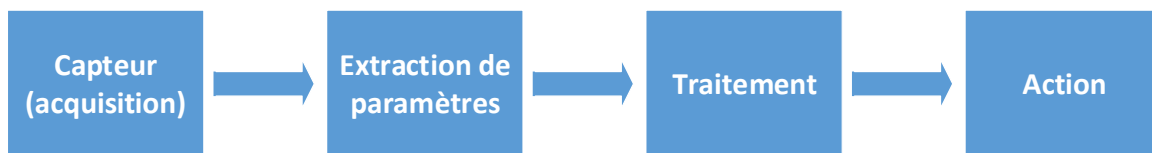


Figure 4.1 Schéma bloc du système

4.1 Capteur

Tout d'abord, portons notre attention sur le capteur utilisé. En effet, comme la question porte autour de l'utilisation d'une caméra thermique basse résolution et que le modèle a été fixé avant le début des travaux, il est utile d'en connaître les caractéristiques de haut niveau.

Il s'agit de la caméra MLX90620 de Melexis (figure 4.2).

¹Figure tirée de la documentation officielle du capteur MLX90620 de Melexis

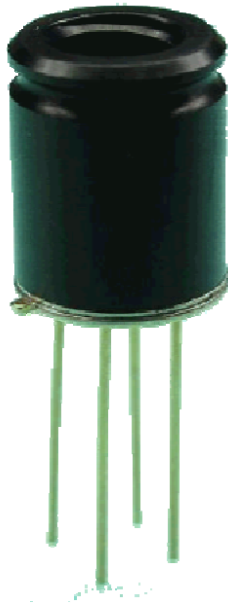


Figure 4.2 Caméra thermique basse résolution MLX90620¹

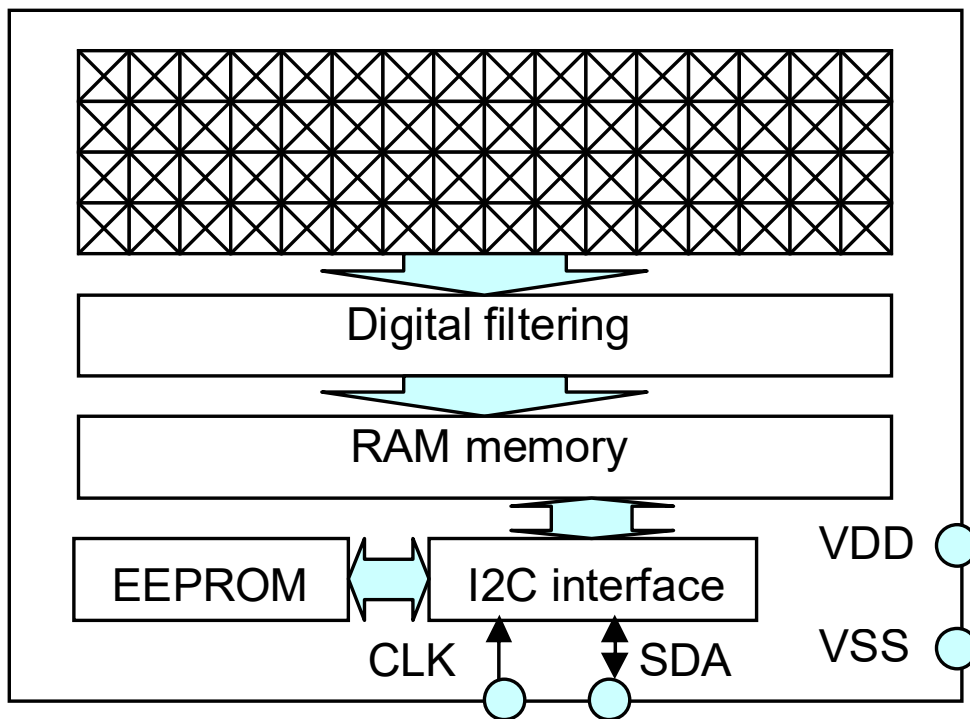


Figure 4.3 Diagramme fonctionnel du MLX90620¹

Cette caméra permet d'obtenir une grille de 4 par 16 mesures thermiques en plus de la température ambiante. À la figure 4.3, on peut voir une représentation de cette grille qui est constituée d'une matrice de thermopiles.

Ces thermopiles sont en fait constituées de thermocouples assemblés en série qui mesurent leur échauffement suite à l'exposition au rayonnement infrarouge (IR) provenant d'un objet distant.

Comme le fonctionnement d'un thermocouple repose l'effet Seebeck (utilisation de fils de métaux différents dont les jonctions sont exposées à des températures différentes) [Reed, 2014] et que l'une de ces jonctions est exposée au IR, nous avons besoin d'une autre mesure : l'autre jonction qui sera mesurée comme étant la température ambiante du capteur (dénotée T_a dans sa documentation).

Comme une différence de température entre ces deux jonctions ne produit qu'un très petit voltage (de l'ordre du microvolt), il est utile de noter que le capteur possède un amplificateur ainsi qu'un convertisseur analogue à digital (ADC) pour chaque pixel.

Les valeurs numérisées seront ensuite conservées en mémoire (RAM sur la figure 4.3) en attendant d'être récupérées via l'interface I_2C par notre prototype.

À la section 5.2.1, plus de détails sur les équations impliquées pour obtenir la température de l'objet distant (TO dans la documentation) seront fournis.

4.2 Extraction de paramètres

Une fois la température acquise, il est nécessaire de pouvoir l'interpréter. Selon la méthode de traitement utilisé (deux méthodes sont décrites à la section 4.3), certains paramètres s'avèrent utiles, voire nécessaires. Il s'agit donc d'une méthode pour réduire l'information en un format utilisable par les algorithmes de traitement.

En effet, même si les expérimentations ont révélé que seuls quelques paramètres étaient nécessaires, il a été utile d'en extraire plus pour analyser lesquels jouent un rôle principal pour la détection de situations dangereuses.

Tout d'abord, considérons que puisque les informations de température sont acquises à l'aide d'une caméra thermique que nous avons une image thermique. Ainsi, chaque point de mesure peut être considéré comme étant un pixel.

Ensuite, il est possible d'obtenir des paramètres relativement simples (moyenne, maximum, minimum, etc.) ou plus complexes. Par exemple, il est possible d'utiliser le temps ou la géométrie pour obtenir une information plus détaillée sur la situation.

Au niveau temporel, il est possible de filtrer les images (filtre passe-bas) pour en éliminer le bruit. Ce filtre passe-bas se concrétise par une moyenne sur n images successives d'un autre paramètre. Par exemple, pour déterminer la valeur de température maximale une fois l'image i atteinte sur un nombre d' n images, la formule 4.1 sera utilisée.

$$max_filtre_i = \sum_{x=i-n}^i \frac{max_i}{n} \quad (4.1)$$

Il est également possible de suivre l'évolution dans le temps. Il suffit de prendre une valeur filtrée et de faire la différence (delta (δ)) avec la valeur filtrée de l'image précédente.

Ainsi, faire suite à l'exemple de l'équation 4.1, pour déterminer $delta_max_filtre_i$, nous utiliserons l'équation 4.2.

$$delta_max_filtre_i = max_filtre_i - max_filtre_{i-1} \quad (4.2)$$

Pour la géométrie, si l'on considère tous les pixels au-dessus d'un certain seuil de température (déterminé dans un fichier de configuration, voir l'annexe A) comme étant « chaud », on peut alors tenter de voir si ceux-ci forment une zone continue (pixels voisins, tel qu'illustré à la figure 4.4 où les pixels en bleu sont voisin du pixel en rouge).

1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64

Figure 4.4 Pixels étant considérés voisins (en bleu) du pixel 6 (en rouge)

Si un tel regroupement de pixels voisins existe, celui-ci sera appelé « blob » (figure 4.5). On note d'ailleurs que plus d'un « blob » peut être présent sur une même image. Par exemple, il peut s'agir d'un rond de la cuisinière ou d'un ustensile de cuisson chaud.



Figure 4.5 Exemple de blob (entouré de vert) lorsque le seuil de température est de 75°C

Il est également possible de combiner l'extraction de paramètres simples au filtrage et la détection de « blobs » pour obtenir des paramètres complexes tels qu'un delta (δ) du maximum de température de tous les « blobs » détectés.

La liste complète des paramètres extraits accompagnés de leur description se trouve en annexe B.

4.3 Traitement

Une fois les données acquises, il reste à déterminer si la situation captée est dangereuse et quantifier ce niveau de danger. C'est à cette étape qu'intervient le traitement à l'aide d'intelligence artificielle.

4.3.1 Logique floue

La première approche qui a été exploitée est l'utilisation de logique floue.

Cette approche consiste à utiliser un ensemble de règles pour reproduire le raisonnement d'un expert [Negnevitsky, 2005]. Il n'y a donc pas d'intelligence artificielle à proprement parler, mais plutôt l'application par le système d'une intelligence externe.

Diverses techniques ont été mises en œuvre, passant de la simple intuition à l'analyse de courbes en passant par la création de diagrammes de dispersions pour tenter de trouver une combinaison de paramètres permettant de détecter une situation dangereuse.

Après plusieurs tentatives, une solution viable a été trouvée en utilisant principalement deux paramètres pour déterminer le niveau de danger, principalement en lien avec les brûlures. Il s'agit de « `delta_blobsGlobalMin` » et de « `delta_min` » (voir l'Annexe B pour les détails). Rappelons que le premier contient la variation du minimum d'une zone dépassant un seuil et le dernier la variation du minimum de toute l'image thermique (le nombre d'images et le seuil utilisé sont déterminés dans la configuration).

Cette configuration s'avère efficace pour détecter un risque élevé de brûlure puisque lorsque ces deux paramètres varient en sens opposés, il y a occlusion. Il peut s'agir d'une main qui passe entre la caméra et la surface de cuisson chaude ou d'un ustensile de cuisson (généralement une main sera à proximité).

Cette détection est possible puisqu'une occlusion au-dessus d'un rond chaud cache la partie supérieure d'un « blob ». Ceci a pour effet d'abaisser rapidement la température des pixels concernés, les excluant de la zone chaude. Ces pixels étant exclus du « blob », celui-ci voit son minimum augmenter (ses pixels les plus loin de la source de chaleur, avec les températures les plus basses, ayant été exclus, il ne reste que ceux plus près du rond). En même temps, comme l'occlusion apporte des éléments plus frais (température de la main ou d'un ustensile froid) vers la surface de cuisson, une baisse de température a lieu sur l'ensemble de la surface de cuisson. Cette baisse affecte entre autres la température minimale.

L'inverse est également vrai : le retrait de l'occlusion rend au « blob » ses pixels un peu plus frais, abaissant son minimum, tout en augmentant la température minimum sur toute la surface de cuisson.

Il est ainsi possible de détecter un mouvement au-dessus d'un rond chaud.

Pour ce qui est de la température détectée sur la surface de cuisson, on considère que si elle dépasse un seuil « normal », il y a danger. En effet, une température au-dessus de 45°C peut causer des brûlures [Wartzek *et al.*, 2011], mais il est fréquent et normal qu'en cuisinant la température d'un ustensile dépasse ce seuil. Il suffit de penser à porter de l'eau à ébullition pour se rendre compte qu'une température avoisinant les 100°C est normale. Cependant, le danger est tout de même présent.

Aussi, la configuration ayant été utilisée pour permettre à l'approche à base de logique floue de quantifier le danger d'une situation utilise la température maximale filtrée sur quelques images pour activer les fonctions d'appartenances autres que l'extrême.

Les figures suivantes permettent de visualiser les outils en place pour aider au développement en plus de donner un exemple de résultat. Notons que les images sont acquises de façon séquentielle à une fréquence de 2Hz dans les expérimentations et numérotées séquentiellement. Ainsi, l'image #1 correspond à la première image d'un enregistrement.

Par exemple, la figure 4.6 donne un exemple des diagrammes de dispersions utilisés pour visualiser les données en vue de déterminer si une combinaison de paramètre pourrait être utile. Dans cette figure, la couleur indique le niveau de danger évalué manuellement (c'est à dire par le développeur, de façon subjective) de façon linéaire (toute valeur entre 0 et 1 possible) (note : cette évaluation a par la suite été transformée pour devenir binaire).

Ainsi, les points les plus intéressants étaient ceux dont la couleur se rapproche le plus du rouge (danger) et ensuite les bleus (sécurité) pour définir les règles à utiliser avec la logique floue.

Aussi, les figures 4.7 et 4.8 nous offrent une situation en tout début d'enregistrement (image #1) sans dangers apparents.

La figure 4.7 nous montre l'interface développée pour visualiser les données. On y voit donc les valeurs des pixels de température en haut ainsi que l'image de la caméra vidéo.

La figure 4.8 quant à elle nous offre quatre graphiques générés automatiquement par l'outil développé. La première section (Filtered Max Temp, pour reprendre le nom des paramètres extraits) représente graphiquement les quatre fonctions d'appartenance en lien

avec la température maximale filtrée de l'image. On y voit également une barre verticale correspondant à la valeur de « Filtered Max Temp » à ce moment.

De la même façon, la deuxième section (`delta_blobsGlobalMin`) et la troisième section (`delta_min`) représentent par une barre verticale la valeur du paramètre superposé aux trois fonctions d'appartenances du paramètre.

La quatrième section est légèrement différente puisqu'il s'agit de la sortie du système lorsqu'il utilise la logique floue. Chacune des cinq fonctions d'appartenances représentant un niveau de danger pouvant être activé indépendamment des autres selon les règles du système sont représentées par une zone en pointillé lorsqu'inactive ou par une zone pleine sinon. La barre noire verticale représente la valeur unique de sortie générée en utilisant le niveau d'activation de toutes les fonctions d'appartenances.

Ensuite, les figures 4.9 et 4.10 représente l'image #18 du même enregistrement : le rond est laissé sans ustensile et sa température augmente. Le niveau de danger détecté augmente lui aussi.

Les quatre prochaines figures devraient être analysées ensemble puisqu'il y a déplacement au-dessus du rond allumé. Comme le mouvement ne peut se constater que par une succession d'images, les figures 4.11 et 4.12 sont de l'image #72 et les figures 4.13 et 4.14 le l'image #73. On constate donc un retard de 0.5s dans la détection causé par la fréquence d'acquisition et le besoin d'une succession d'images, ce qui est acceptable étant donné que l'on doit détecter un changement, donc utiliser quelques images. Par contre, on constate également que l'occlusion est bel et bien détectée et que le niveau de danger plafonne à ce moment.

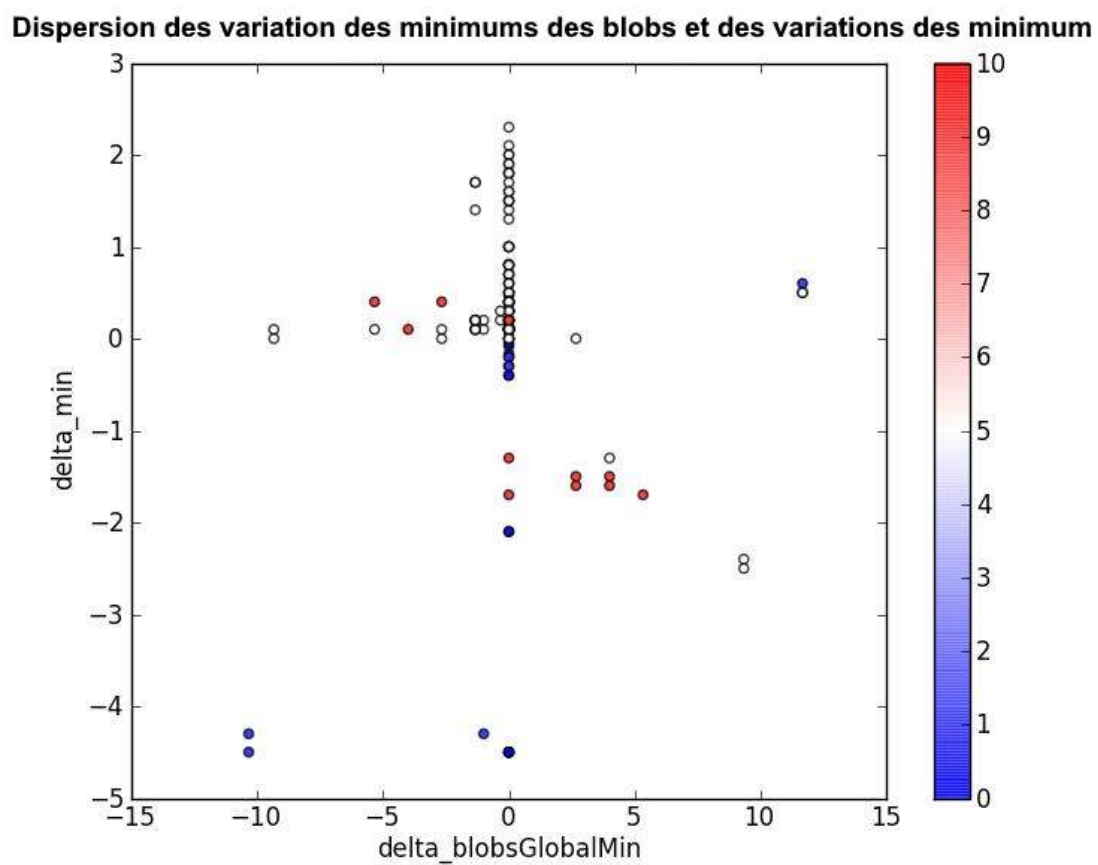


Figure 4.6 Diagramme de dispersion $\text{delta_blobsGlobalMin} - \text{delta_min}$

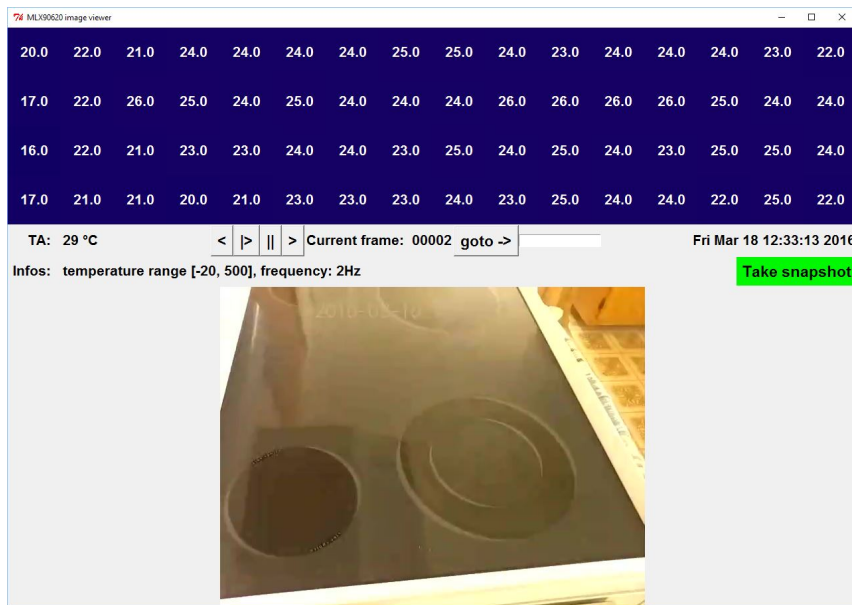


Figure 4.7 Exemple de situation sécuritaire - interface

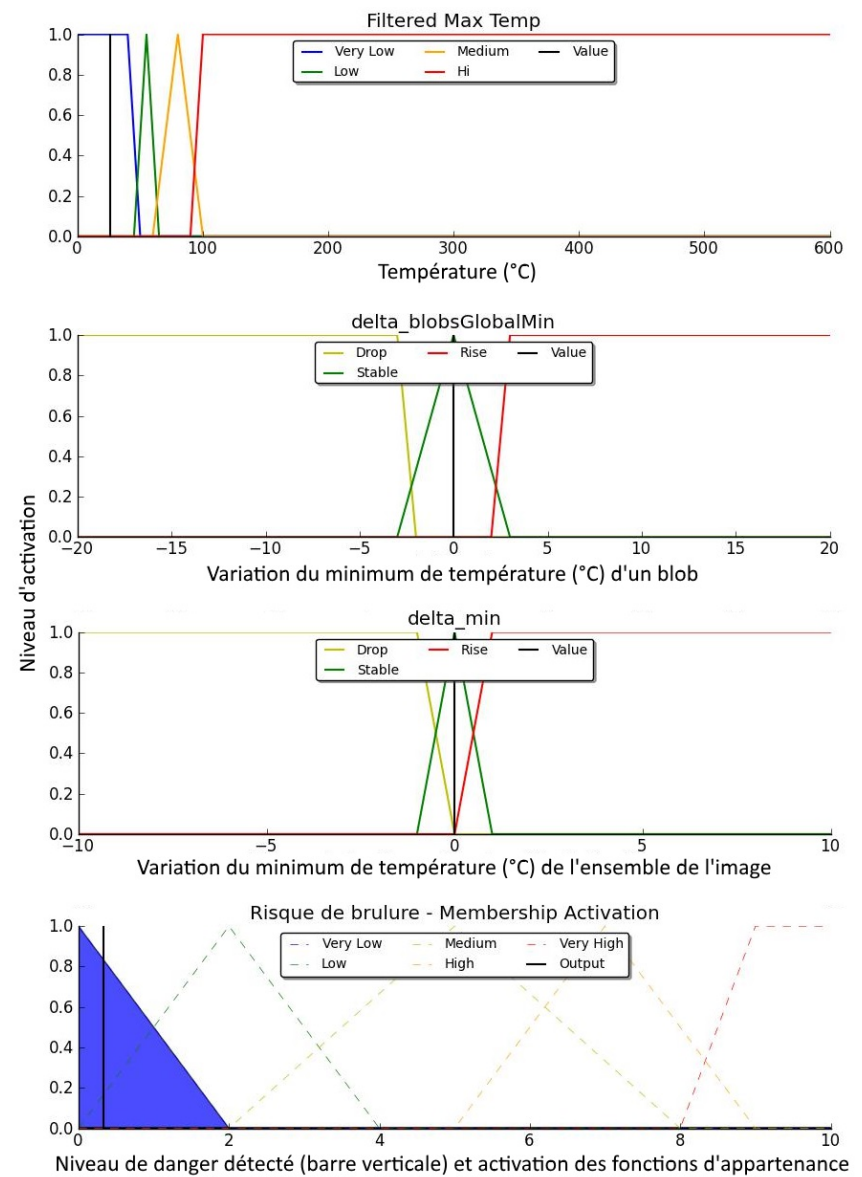


Figure 4.8 Exemple de situation sécuritaire - logique floue

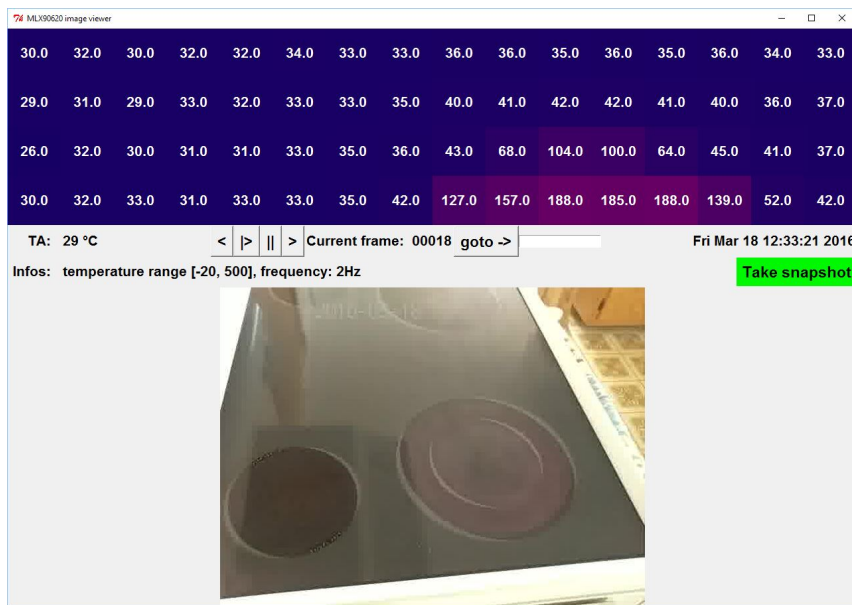


Figure 4.9 Exemple de situation avec rond exposé - interface

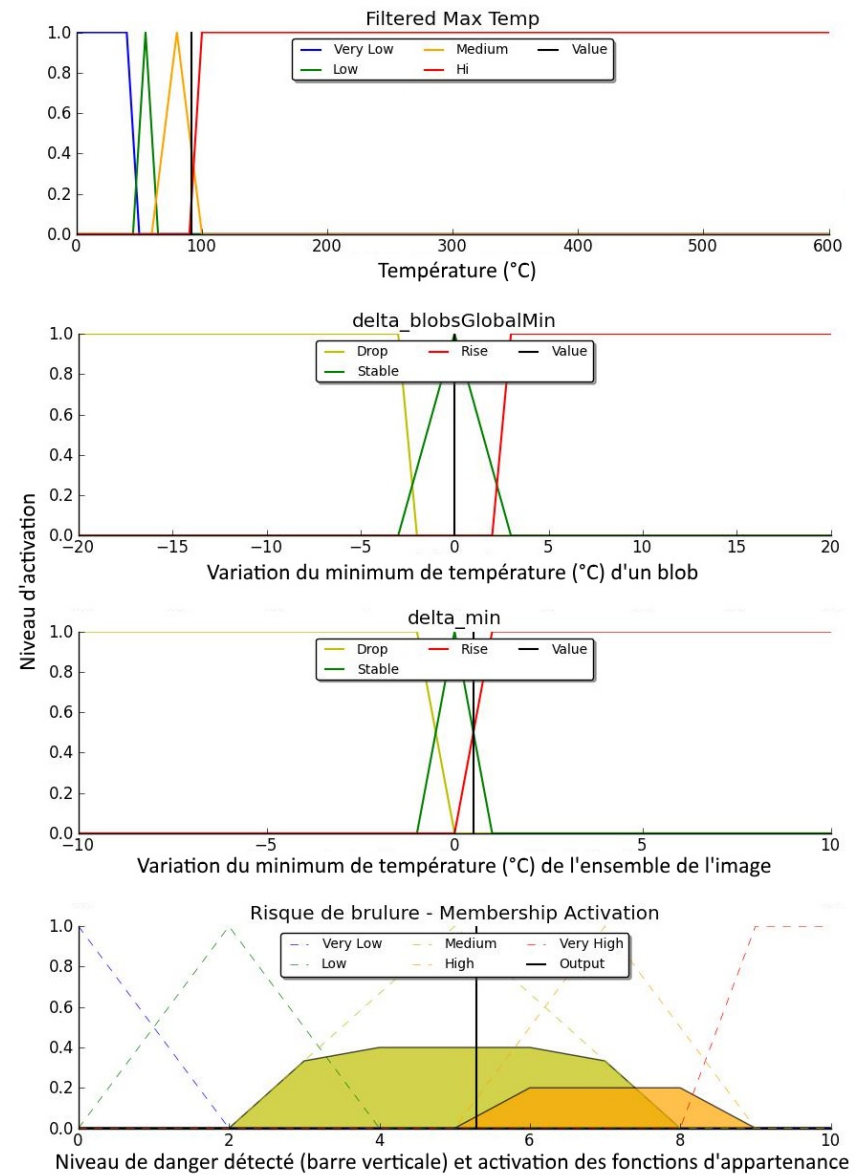


Figure 4.10 Exemple de situation avec rond exposé - logique floue



Figure 4.11 Exemple de situation avec début d'occlusion - interface

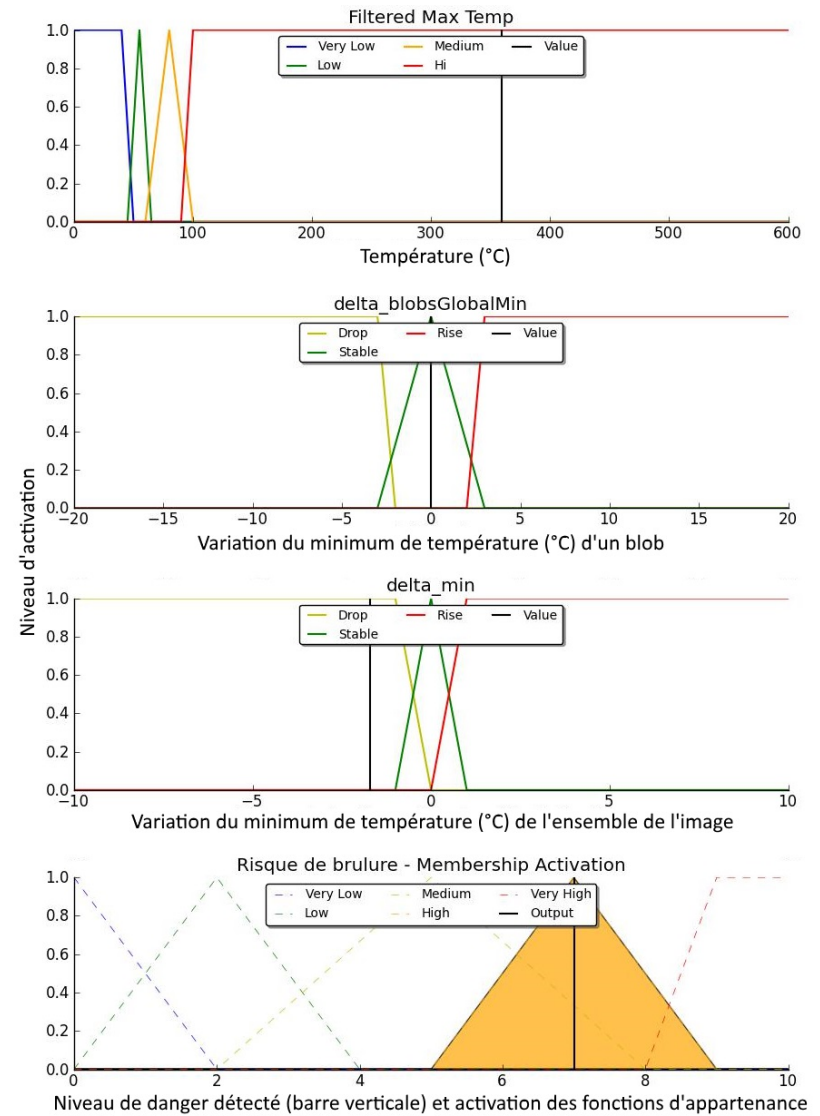


Figure 4.12 Exemple de situation avec début d'occlusion - logique floue

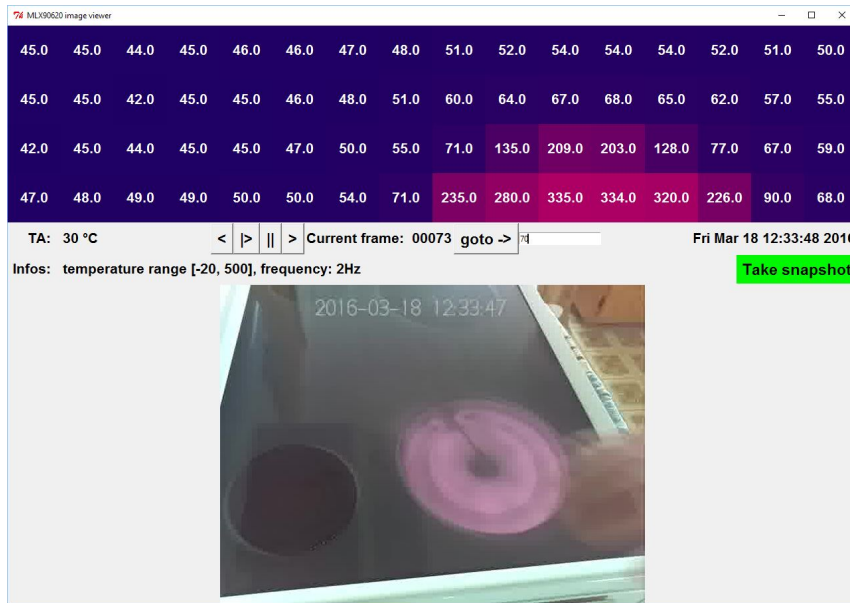


Figure 4.13 Exemple de situation avec occlusion détectée - interface

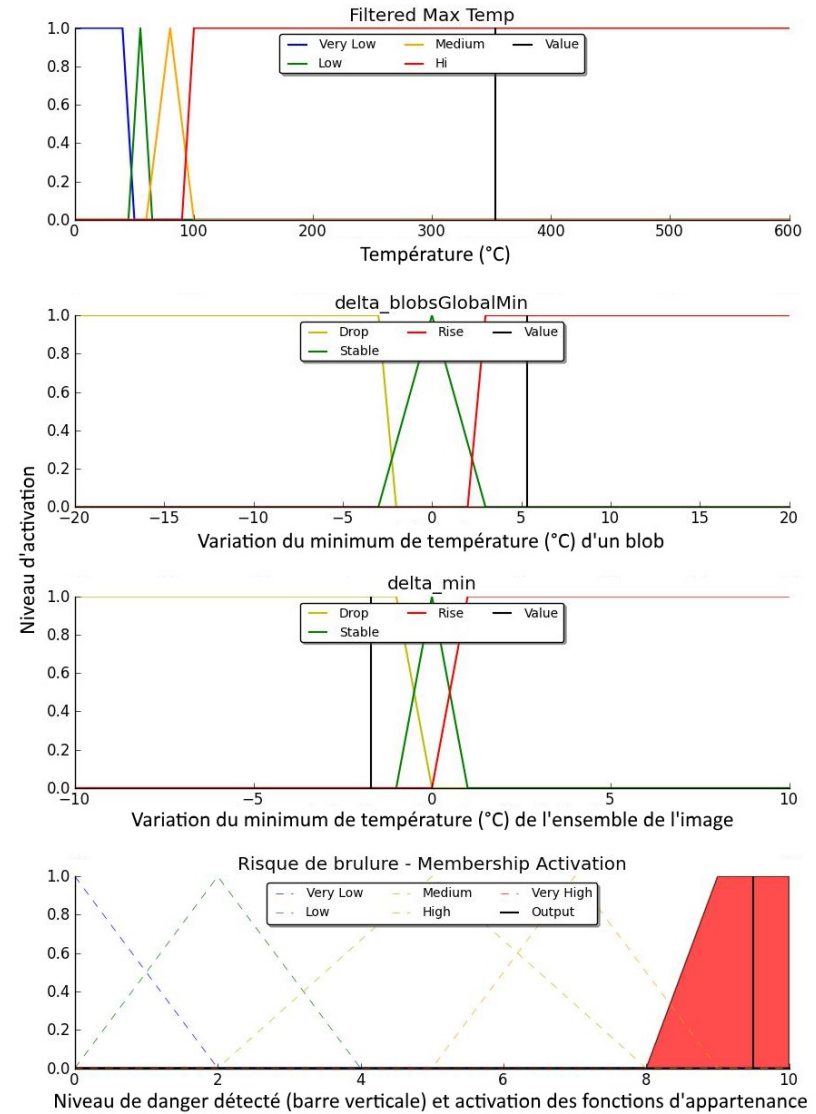


Figure 4.14 Exemple de situation avec occlusion détectée - logique floue

4.3.2 Réseau de neurones

Par la suite, une deuxième approche a été mise de l'avant : l'utilisation d'un réseau de neurones simple.

Dans cette approche, l'extraction de paramètres est peu utile puisque l'on utilise les valeurs des pixels de température ainsi que la température ambiante. Le réseau de neurones doit donc être un peu plus volumineux pour arriver à extraire par lui-même les paramètres qui lui seront utiles [Fischer, 2004]. Cependant, comme il s'agit d'une image de petite taille (64 pixels), la taille du réseau reste relativement petite.

Par contre, les valeurs de température ne peuvent être utilisées directement, elles doivent d'abord être normalisées [Larochelle, 2014; Welch, 2015-2016].

Reste ensuite à déterminer la structure du réseau à utiliser en considérant le type de fonction que celui-ci aura à inférer.

Pour permettre au réseau de représenter une fonction non linéaire, nous utiliserons au moins une couche cachée [Larochelle, 2014; Negnevitsky, 2005; Russell *et al.*, 2010] (donc trois couches et plus).

Les architectures des réseaux évaluées ont variées de trois à cinq couches, selon les tests utilisés, avec autant d'entrées que de points de mesure et une seule sortie. L'une des configuration qui a donné de bons résultats consiste en 65 entrées, trois couches cachées de dix neurones menant à un neurone en sortie (voir figure 4.15).

Pour bien comprendre comment ce réseau fonctionne, commençons par le rôle d'un neurone : combiner ses entrées pondérées et appliquer une fonction non linéaire (figure 4.16 et équation 4.3 où x_i représente les entrées et w_i les poids).

$$o_j = f\left(\sum_{i=1}^n x_i w_i\right) \quad (4.3)$$

Comme le réseau mentionné plus haut comporte plus d'une couche et plusieurs neurones par couches, en calculer la sortie devient plus complexe. Heureusement pour nous, l'utilisation de matrice vient grandement simplifier les équations et limiter le temps de calcul.

En effet, on peut représenter toutes les entrées x_i à l'aide d'une matrice X de n colonnes. Dans notre cas, on parle de 65 colonnes.

²Figure éditée dont l'original se trouve à l'adresse https://fr.wikipedia.org/wiki/Reseau_de_neurones_artificiels

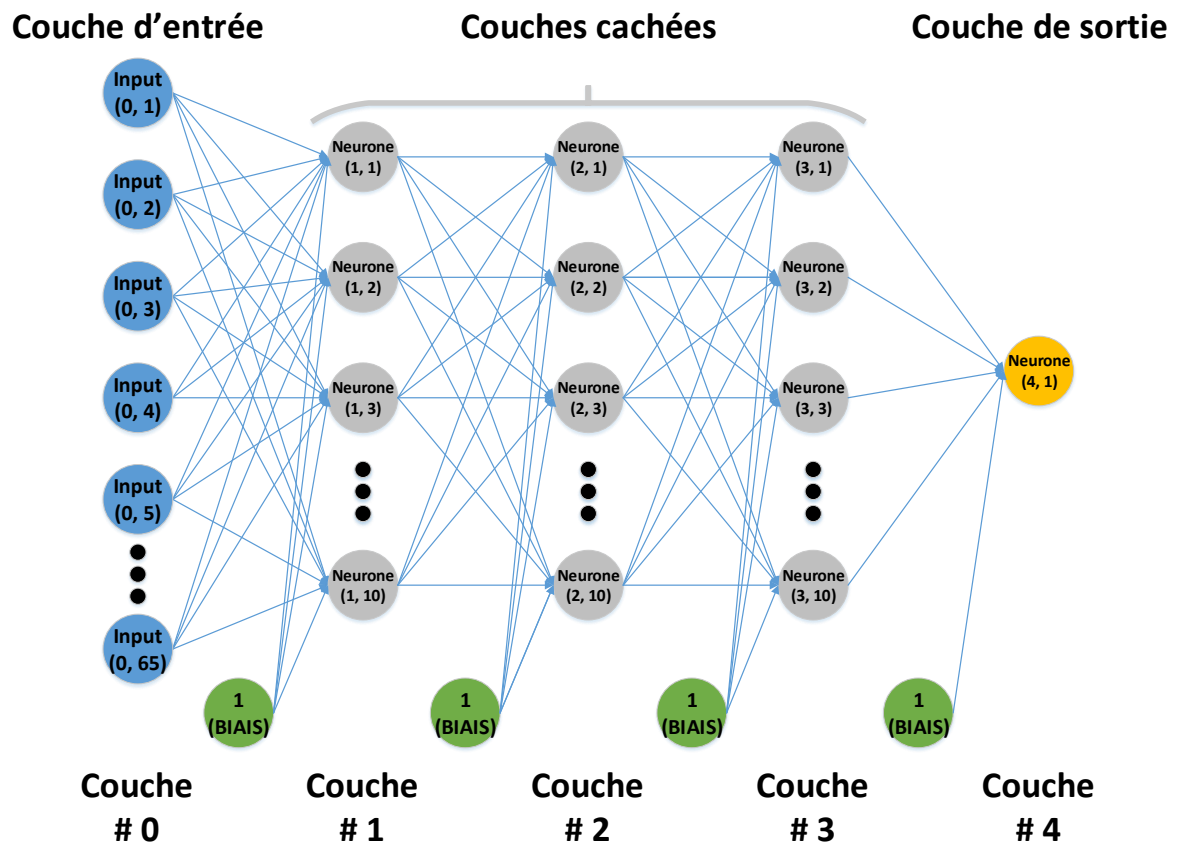
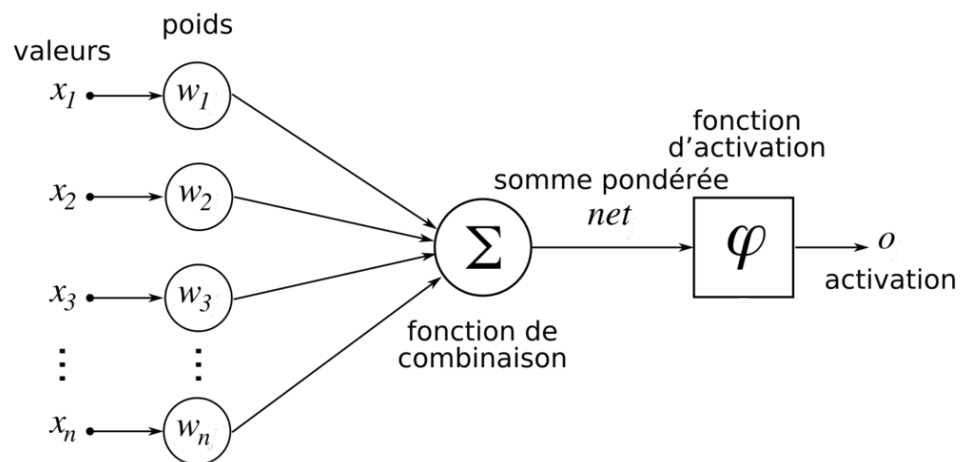


Figure 4.15 Configuration du réseau de neurones

Figure 4.16 Fonctionnement d'un neurone²

Ensuite, considérant qu'à chaque flèche (synapse) de la figure 4.15 intervient un des w_{ij} de la figure 4.16, on peut les grouper en une matrice W comportant autant de lignes que d'entrées et autant de colonnes que de neurones de la couche suivante. On note également la présence de biais [Larochelle, 2014; Russell *et al.*, 2010], dont la valeur est fixée arbitrairement à 1 et est traité comme une entrée pour la couche suivante. On associe donc un poids aux synapses le reliant aux neurones de la couche suivante, poids qui lors de l'entraînement pourra changer, modifiant également la valeur effective de ce biais dans les calculs.

Ainsi, pour le réseau évoqué à la figure 4.15, nous aurons quatre matrices W . La première W_0 (entre les couches 0 et 1), aura 66 lignes (65 entrées et un biais) et 10 colonnes, les matrices W_1 et W_2 auront 11 lignes et 10 colonnes et enfin, la W_3 aura 11 lignes et une seule colonne.

Nous pouvons ainsi transformer l'équation 4.3 avec la notation matricielle pour l'adapter au réseau de la figure 4.15 pour obtenir les équations 4.4 à 4.7.

On remarque au passage l'apparition de nouvelles matrices. Les matrices A_1 à A_3 sont des matrices intermédiaires contenant les activations des couches cachées. On entend par activation le résultat de l'équation 4.3. De plus, à l'équation 4.7 O fait son apparition pour représente les sorties (donc une matrice 1 par 1 dans notre cas).

$$A_1 = f(XW_0) \tag{4.4}$$

$$A_2 = f(A_1W_1) \tag{4.5}$$

$$A_3 = f(A_2W_2) \tag{4.6}$$

$$O = f(A_3W_3) \tag{4.7}$$

Nous avons ainsi, en utilisant successivement les équations 4.4 à 4.7, la possibilité d'obtenir un niveau de danger estimé par une valeur variant entre 0 (sécuritaire) et 1 (danger maximum) à partir des informations du capteur, préalablement normalisées pour que celles-ci soient contenues entre 0 et 1 également.

Cependant, le réseau n'est toujours pas prêt à l'utilisation : une phase d'apprentissage est nécessaire. En effet, la puissance d'un réseau de neurones repose sur sa capacité, à partir de données d'entraînement associées à un résultat désiré, d'extrapoler une sortie pour toute autre situation. Si cet entraînement est bien réalisé, c'est à dire avec des cas sans ambiguïtés, qui représentent bien les situations possibles à évaluer et en quantité suffisante, il sera possible au réseau de bien estimer le niveau de danger d'une situation qui ne lui avait pas été présenté auparavant.

Pour ce faire, nous utiliserons une rétropropagation du gradient qui consiste à déterminer l'erreur entre la prédiction du réseau et la valeur désirée, le tout pendant l'entraînement, pour ensuite ajuster le réseau pour minimiser l'erreur. Bien que conceptuellement simple, le processus se complexifie lorsqu'il y a plusieurs neurones impliqués et plusieurs couches. Ainsi, il faut déterminer la contribution de chaque neurone de la couche $l - 1$ à l'erreur déterminée à la couche l . Par exemple, pour la couche de sortie ($l = 4$), l'erreur est simplement la différence entre la sortie du réseau et la valeur objectif. On doit ensuite, pour chacun des 10 neurones de la couche $l = 3$, présents à la figure 4.15, déterminer quelle est sa contribution à cette erreur, le tout en considérant les poids associés aux synapses (liens entre les neurones) qui les relie à la sortie.

Il est alors possible d'ajuster les poids (matrice W) associés aux synapses en fonction de leur implication, le tout pour minimiser l'erreur. Ensuite, comme d'autres couches précèdent et qu'elles ont elles aussi participé à cette erreur, il convient de continuer à propager la participation à l'erreur vers les couches antérieures.

Concrètement, nous utiliserons l'erreur quadratique [Negnevitsky, 2005; Welch, 2015-2016] entre les valeurs objectifs (Y) et les sorties du système (O) pour déterminer le coût (ou erreur) (J) tel qu'indiqué à l'équation 4.8. De plus, l'introduction d'une division par deux facilitera le travail pour dériver l'équation dans les étapes subséquentes.

$$J = \sum \frac{1}{2}(Y - O)^2 \quad (4.8)$$

Aussi, même si le réseau ne produit en sortie qu'une matrice d'une seule ligne et une seule colonne lorsqu'on l'utilise, il en est autrement lors de l'entraînement. En effet, l'utilisation de matrice permet d'utiliser plusieurs images (une par ligne de matrice) en entrée pour produire en un seul passage plusieurs résultats. Ce comportement est particulièrement utile pour l'entraînement, mais n'est pas utilisé pour déterminer un niveau de danger en temps réel puisque les images arrivent une à la fois dans ce cas. Ainsi, J , utile uniquement pour

l'entraînement, représente le coût global, donc l'erreur globale entre les résultats associés à chaque image et leur valeur objectif respectives.

Ensuite, pour minimiser ce coût efficacement, il est important de déterminer le gradient, donc comment dans quelle « direction » faire varier les poids pour minimiser l'erreur. Pour ce faire, si l'on combine les équations 4.4 à 4.7 et 4.8, nous obtenons l'équation 4.9. Cette dernière nous permet de constater que le coût peut varier en fonction des entrées (X), des poids (W) et de l'objectif pour l'entraînement (Y). Par contre, l'entraînement n'a pas la possibilité de modifier les entrées (X) ou sorties désirées associées (Y). Nous devons donc modifier les poids (W) pour procéder à la minimisation du coût, donc l'entraînement du réseau.

$$J = \sum \frac{1}{2} (Y - f(f(f(f(XW_0)W_1)W_2)W_3))^2 \quad (4.9)$$

Il nous faut donc un gradient particulier : celui associé à la dérivée partielle en fonction des différents W . Commençons par la matrice de poids la plus près de la couche de sortie (W_3) (équation 4.10).

$$\frac{\partial J}{\partial W_3} = \frac{\partial \sum \frac{1}{2} (Y - O)^2}{\partial W_3} \quad (4.10)$$

Puisque la dérivée d'une somme correspond à la somme des dérivés, nous pouvons passer de 4.10 vers 4.11.

$$\frac{\partial J}{\partial W_3} = \sum \frac{\partial \frac{1}{2} (Y - O)^2}{\partial W_3} \quad (4.11)$$

Ensuite, l'application de la règle des exposants, combiné à notre division par 2, nous permet d'éliminer l'exposant et avec l'application du théorème de dérivation des fonctions composées, nous pouvons passer aux équations 4.12.

$$\frac{\partial J}{\partial W_3} = \sum (Y - O) \cdot -\frac{\partial O}{\partial W_3} \quad (4.12a)$$

$$= \sum (Y - O) \cdot -\frac{\partial O}{\partial A_3 W_3} \frac{\partial A_3 W_3}{\partial W_3} \quad (4.12b)$$

À ce moment, il devient essentiel de connaître f , la fonction d'activation utilisée à l'intérieur des neurones, pour ensuite évaluer $\frac{\partial O}{\partial A_3 W_3}$. Il s'agit d'une fonction sigmoïde (équation 4.13).

$$f(z) = \frac{1}{1 + e^{-z}} \quad (4.13)$$

Nous pouvons ensuite dériver (équation 4.14) cette fonction pour continuer la dérivée partielle de l'équation 4.12 vers 4.15.

$$f'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} \quad (4.14)$$

$$\frac{\partial J}{\partial W_3} = \sum (Y - O) \cdot -f'(A_3 W_3) \frac{\partial A_3 W_3}{\partial W_3} \quad (4.15)$$

Question de garder les équations simples, substituons une partie de l'équation par une nouvelle variable δ_4 (équation 4.16) puisqu'un certain patron se répètera lors de la rétro-propagation aux autres couches.

$$\delta_4 = -(Y - O)f'(A_3 W_3) \quad (4.16)$$

Ainsi, 4.15 devient 4.17 à l'aide de 4.16.

$$\frac{\partial J}{\partial W_3} = \sum \delta_4 \frac{\partial A_3 W_3}{\partial W_3} \quad (4.17)$$

Une particularité intéressante des matrices est qu'en effectuant une multiplication, on peut également effectuer la somme des $\frac{\partial J}{\partial W_3}$. Pour ce faire, puisque $\frac{\partial A_3 W_3}{\partial W_3}$ est l'activité de nos synapses et que les valeurs dans δ_4 doivent être multipliées par les activités des synapses, il est à propos de transposer A_3 et faire une multiplication matricielle avec δ_4 . L'évaluation de $\frac{\partial A_3 W_3}{\partial W_3}$ permet donc d'obtenir l'équation 4.18.

$$\frac{\partial J}{\partial W_3} = A_3^T \delta_4 \quad (4.18)$$

Ensuite, il nous faut propager ce gradient vers les autres couches, d'où le terme rétro-propagation.

Les équations 4.19 nous permettent de constater que δ_4 est réutilisable dans le calcul de $\frac{\partial J}{\partial W_2}$.

$$\frac{\partial J}{\partial W_2} = \frac{\partial \sum \frac{1}{2}(Y - O)^2}{\partial W_2} \quad (4.19a)$$

$$= \sum \frac{\partial \frac{1}{2}(Y - O)^2}{\partial W_2} \quad (4.19b)$$

$$= \sum (Y - O) \cdot -\frac{\partial O}{\partial W_2} \quad (4.19c)$$

$$= \sum (Y - O) \cdot -\frac{\partial O}{\partial A_3 W_3} \frac{\partial A_3 W_3}{\partial W_2} \quad (4.19d)$$

$$= \sum (Y - O) \cdot -f'(A_3 W_3) \frac{\partial A_3 W_3}{\partial W_2} \quad (4.19e)$$

$$= \sum \delta_4 \frac{\partial A_3 W_3}{\partial W_2} \quad (4.19f)$$

Maintenant, poursuivons pour propager à la couche inférieure en utilisant une fois de plus le théorème de dérivation des fonctions composées pour passer de 4.19 à 4.20. Notons que pour alléger la notation et puisque l'on sait que la multiplication matricielle s'occupera de la somme, celle-ci sera abandonnée à partir des équations 4.20.

$$\frac{\partial J}{\partial W_2} = \delta_4 \frac{\partial A_3 W_3}{\partial A_3} \frac{\partial A_3}{\partial W_2} \quad (4.20a)$$

$$= \delta_4 W_3^T \frac{\partial A_3}{\partial W_2} \quad (4.20b)$$

$$= \delta_4 W_3^T \frac{\partial A_3}{\partial A_2 W_2} \frac{\partial A_2 W_2}{\partial W_2} \quad (4.20c)$$

$$= \delta_4 W_3^T f'(A_2 W_2) \frac{\partial A_2 W_2}{\partial W_2} \quad (4.20d)$$

$$= A_2^T \delta_4 W_3^T f'(A_2 W_2) \quad (4.20e)$$

Ainsi, on utilise à quelques reprises le théorème de dérivation des fonctions composées et on reproduit un patron déjà vu lors du calcul de $\frac{\partial J}{\partial W_3}$. En comparant l'équation 4.20d à 4.15, on constate qu'il y a quelques termes qui sont multipliés par le résultat d'un appel à f' pour ensuite être multiplié à une dérivée partielle. Si on reproduit l'extraction de δ_4 , il

est possible d'extraire un δ_3 (équation 4.21) pour alléger l'écriture de l'équation 4.20d et nous donne 4.22a (similaire à 4.17).

$$\delta_3 = \delta_4 W_3^T f'(A_2 W_2) \quad (4.21)$$

$$\frac{\partial J}{\partial W_2} = \delta_3 \frac{\partial A_2 W_2}{\partial W_2} \quad (4.22a)$$

$$= A_2^T \delta_3 \quad (4.22b)$$

On remarque alors que l'équation 4.22b ressemble à l'équation 4.18. Il en sera de même pour les autres couches.

Ainsi, nous pourrions calculer $\frac{\partial J}{\partial W_1}$ (équations 4.24) et $\frac{\partial J}{\partial W_0}$ (équations 4.25) de la même façon.

Il suffit alors de reproduire le même patron d'utiliser le δ de la couche précédente, le multiplier au W transposé qui relie la couche précédente à la couche actuelle et au résultat de l'appel de f' sur le AW qui permet de calculer l'activation de la couche actuelle pour obtenir le nouveau δ . Le tout est exprimé mathématiquement à l'équation 4.23 où l représente la couche actuelle. On peut d'ailleurs considérer l'équation 4.21 comme étant un exemple pour 4.23.

$$\delta_l = \delta_{l+1} W_l^T f'(A_{l-1} W_{l-1}) \quad (4.23)$$

$$\delta_2 = \delta_3 W_2^T f'(A_1 W_1) \quad (4.24a)$$

$$\frac{\partial J}{\partial W_1} = A_1^T \delta_2 \quad (4.24b)$$

$$\delta_1 = \delta_2 W_1^T f'(A_0 W_0) \quad (4.25a)$$

$$\frac{\partial J}{\partial W_0} = A_0^T \delta_1 \quad (4.25b)$$

Un autre aspect disponible avec l'utilisation d'un réseau de neurones consiste à introduire un paramètre pour limiter le surapprentissage [Russell *et al.*, 2010; Welch, 2015-2016]. Cette technique n'a toutefois pas été retenue puisque les résultats étaient déjà intéressants.

4.4 Action

Lorsque le niveau de danger est quantifié, il reste à produire une action.

Ce qui a été envisagé est d'offrir une action informative à l'utilisateur, principalement à l'aide d'une lampe dont l'intensité et/ou la couleur pourraient varier en fonction du danger. Il est également possible d'offrir une information plus axée vers la prévention et l'aide à l'encadrement d'une personne en perte d'autonomie en fournissant un journal aux intervenants ou aidants. Ce journal pourrait regrouper la nature et la fréquence des dangers détectés et ainsi permettre d'évaluer la sécurité de laisser accès à la cuisinière. Ce type d'aide aux intervenants pourrait même être poussé jusqu'à l'envoi d'alerte en temps réel (message texte ou courriel) en cas de besoin.

Nous avons également écarté une action potentielle : couper l'alimentation électrique de la cuisinière en cas de danger jugé extrême puisque cette approche comporte plusieurs inconvénients :

- augmentation de la complexité de la solution en nécessitant l'utilisation de matériel électrique plus complexe (installation) et plus coûteux (ce qui peut éventuellement nuire à l'accessibilité d'un produit)
- risque de frustration de l'utilisateur, menant à un contournement du système (risque augmenté en cas de faux positif ou même si le système est parfaitement fonctionnel puisque l'utilisateur peut être une personne qui n'a pas la capacité de juger du niveau de danger de la situation)

En ce qui concerne les actions qui ont été retenues, l'utilisation d'une D.E.L. dont on peut faire varier la couleur pourrait permettre de donner une rétroaction immédiate et intuitive à l'utilisateur. En effet, une situation sécuritaire pourrait produire une lumière verte qui passerait graduellement au rouge plus la situation est considérée à risque par le système.

Imaginons que le système soit fixé sous la hotte de la cuisinière ou à proximité et il est alors envisageable de remplacer l'éclairage habituel par ce système (qui pourrait produire un éclairage blanc pour des fins de commodités en plus de la rétroaction). Le système pourrait alors être alimenté en électricité en plus d'être positionné pour bien « voir » la surface de cuisson.

Il a également été considéré que la création d'un journal accessible aux aidants par un portail web ou de l'informatique nuagique. Ce journal pourrait permettre de répertorier les événements considérés comme à risque par le système en vue de quantifier la fréquence et la gravité de ces événements.

Il pourrait également s'avérer intéressant d'utiliser ce journal pour analyser l'évolution de la gravité et de la fréquence (considérant des personnes en perte d'autonomie) pour déterminer si l'utilisation sans supervision directe d'un aidant est toujours appropriée pour l'utilisateur.

CHAPITRE 5

EXPÉRIMENTATIONS

Les expérimentations reliées à ce projet ont eu lieu dans une véritable cuisine. Il était donc nécessaire de réaliser des expérimentations relativement sécuritaires puisque cet environnement n'était pas équipé pour faire face à des situations pouvant mener à un début d'incendie.

Ainsi, des enregistrements de cuisson réelle ont été réalisés, mais également quelques cas d'utilisation scénarisés ont été réalisés dans le but d'obtenir des données de taille limitée et comportant des imprudences. Il était donc plus facile de travailler avec ces enregistrements condensés pour valider la performance du système, ceux-ci étant de plus courte durée et étant plus intéressants puisqu'ils permettaient de faire varier le niveau de danger que l'on désirait obtenir du système.

Précisions que même si les travaux de recherche utilisent des enregistrements, le système peut aussi bien travailler en temps réel qu'en différé. En fait, les enregistrements étaient rejoués à la vitesse d'acquisition lors des traitements, ce qui confirme que le traitement peut se faire en temps réel.

Aussi, comme le système ordonné de traitement ne possède pas d'interface graphique, il était plus complexe, voir impossible, de visualiser les données sur celui-ci. Les expérimentations se sont donc déroulées à l'aide du prototype pour l'acquisition et la transmission des données et d'un ordinateur distant pour le traitement et la visualisation des résultats.

De plus, comme il était envisagé d'intégrer ce système dans un autre de plus grande ampleur (laboratoire DOMUS), une transmission des données à l'aide du protocole TCP/IP était un atout.

Il est cependant réaliste de présager que le système ordonné, seul, pourrait suffire puisque ce dernier avait la tâche d'acquérir et d'enregistrer les données du capteur, mais également d'enregistrer des images capturées par une caméra vidéo de façon synchronisée.

Bref, la demande étant plus grande pour les besoins expérimentaux, il est probable que le système ordonné soit suffisant en ne demandant qu'une acquisition et un traitement.

5.1 Matériel utilisé

5.1.1 Montage utilisé

Dans le but de faciliter l'acquisition des données, un montage physique (figures 5.1 et 5.2) a été réalisé en incorporant non seulement le système ordonné prévu (Raspberry Pi) et la caméra thermique de Melexis, mais également une caméra vidéo. Celle-ci ayant pour but de permettre d'autodocumenter les enregistrements qui allaient par la suite servir à tester les algorithmes utilisés.

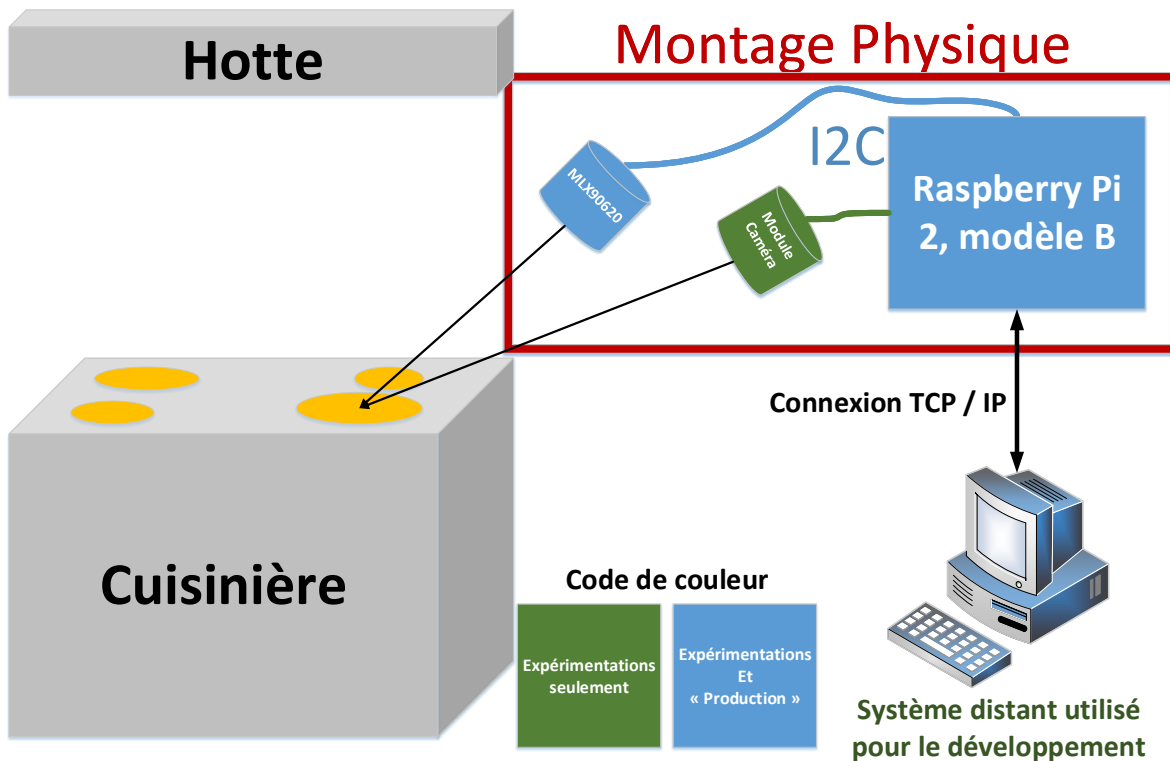


Figure 5.1 Schéma du montage

Le montage consistait en deux sections de bois pouvant se serrer l'une contre l'autre afin de se fixer sur le cache lumière d'armoires de cuisine (figure 5.3). Il était ainsi possible de fixer le prototype sans abimer l'environnement dans lequel les enregistrements ont eu lieu.

Aussi, comme le montre la figure 5.1, les deux caméras (thermique et vidéo) étaient orientées de façon à surveiller un rond en particulier. Il était toutefois possible de « voir » une partie du reste de la surface de cuisson, mais la résolution de la caméra thermique (MLX9620) étant limitée, un choix devait être fait pour le positionnement. La figure 5.10 permet de constater la limitation : la résolution verticale étant uniquement de quatre pixels, il est difficile, sans augmenter la distance entre la cuisinière et le capteur, de sur-

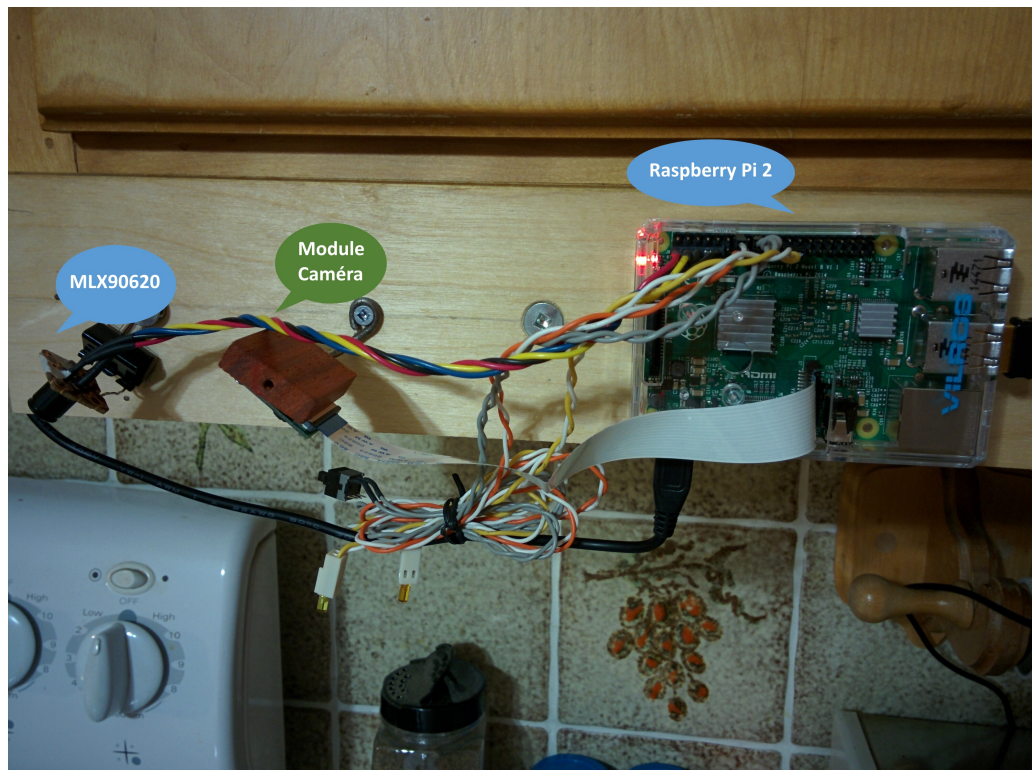


Figure 5.2 Montage servant à l'enregistrement des données

veiller à la fois l'élément chauffant et l'ustensile qui peut y être déposé. Le positionnement utilisé a l'avantage d'être suffisamment près pour limiter les pertes de précisions de mesure tout en permettant de voir environ une moitié de la cuisinière (avant-arrière).

Au sujet de la caméra vidéo utilisée, il s'agit du module caméra spécifique au RaspberryPi[[The Raspberry Pi Foundation, 2017a](#)] (figure 5.6). Elle a été configurée, dans le cadre du projet, pour capter deux images par seconde, donc à la même fréquence que la caméra thermique.

Rappelons que comme la légende de la figure 5.1 le mentionne, cette caméra vidéo n'est présente que pour les fins d'expérimentations. En mode « Production », donc lorsque le prototype effectue la supervision d'une situation non expérimentale, la caméra vidéo ainsi que le système distant ne sont pas utilisés).

On peut également remarquer la présence de D.E.L. et d'un bouton vers le bas du montage. Bien qu'il ne s'agissait pas d'une nécessité, ces ajouts ont été pertinents à rendre l'acquisition de données facile et pratique.

En effet, le prototype a été configuré pour que dès que RaspberryPi est mis en marche, celui-ci débute un enregistrement de façon autonome. Les D.E.L. permettaient à ce moment

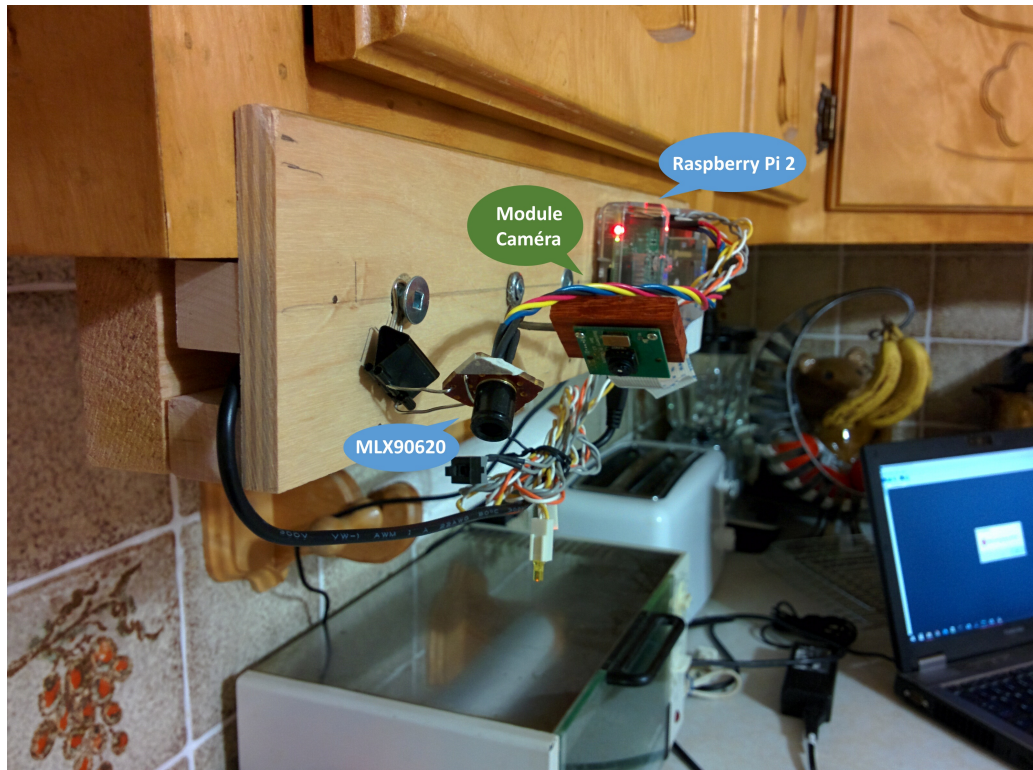


Figure 5.3 Montage servant à l'enregistrement des données, vu de côté

de confirmer que l'enregistrement était bel et bien actif et le bouton permettait de terminer l'enregistrement et démarrer la séquence de mise hors tension du système. Il ne transmet donc aucune donnée et aucun traitement en temps réel n'est effectué.

Ces enregistrements étaient par la suite téléchargés sur un autre système en vue d'y être visualisés et d'éventuellement les utiliser pour tester la détection de danger.

Un schéma du montage est d'ailleurs disponible à la figure 5.1.

Une version plus ancienne du montage, qui illustre un peu mieux un cas d'utilisation réel (sans la caméra vidéo¹) est disponible à la figure 5.4.

¹On peut également remarquer que le système ordonné est un Raspberry Pi de première génération sur cette photo.



Figure 5.4 Montage sans caméra

5.1.2 Système ordonné



Figure 5.5 Raspberry Pi 2, Model B²



Figure 5.6 Module caméra du Raspberry Pi³

Le système ordonné qui a été retenu pour ce projet est le Raspberry Pi 2, modèle B[The Raspberry Pi Foundation, 2017b]. Il a été choisi pour son faible coût, sa disponibilité et pour bénéficier de l'expertise antérieure du développeur sur ce système. Bien entendu, ces raisons n'auraient pas suffi s'il n'offrait pas les performances ou la capacité de s'interfacer avec le capteur.

Contrairement à un système embarqué, il s'agit d'un ordinateur avant tout. Cette caractéristique en fait un atout du point de vue développement logiciel et possibilités de traitement.

Aussi, les limitations imposées par un système utilisant une distribution de Linux au niveau du cadencement ne nous empêchent pas d'avoir un échantillonnage suffisant (2Hz).

Au niveau des caractéristiques de ce système, il se détaille à un prix de 35\$ US et possède un processeur « quad-core ARM Cortex-A7 » cadencé à 900MHz accompagné de 1Go de mémoire vive. Au niveau des ports disponible sur l'appareil, on retrouve : des ports USB (4x), 40 connecteurs d'entrée/sortie à usage général (GPIO), un port HDMI, un port Ethernet, une sortie audio et vidéo combinée en un port de 3.5mm, une interface pour le module caméra (CSI), un port vers un module d'affichage (DSI), une fente pour carte microSD et enfin un port microUSB pour l'alimentation électrique.

²figure disponible à l'adresse suivante : <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

³Source de l'image : <https://cdn.sparkfun.com/assets/parts/8/2/7/8/11868-00a.jpg>

Ce sont d'ailleurs les GPIO qui permettent d'y connecter le capteur et de brancher des D.E.L. et boutons (tel que décrit à la section [5.1.1](#)). L'annexe [C](#) permet de constater la souplesse ainsi que la disponibilité de standards sur certaines broches réservées à cet effet. Il a donc été possible de connecter la caméra thermique MLX90620, présentée à la section [4.1](#), sur les broches 1, 3, 5 et 9 et d'utiliser quelques ports génériques disponibles pour les D.E.L. et bouton discuté à la section [5.1.1](#).

De plus, comme nous avons utilisé le module caméra du Raspberry Pi (figure [5.6](#)) pour acquérir les images correspondantes aux images thermiques, nous avons pu utiliser le port dédié à la caméra sur l'appareil.

5.2 Configuration logicielle

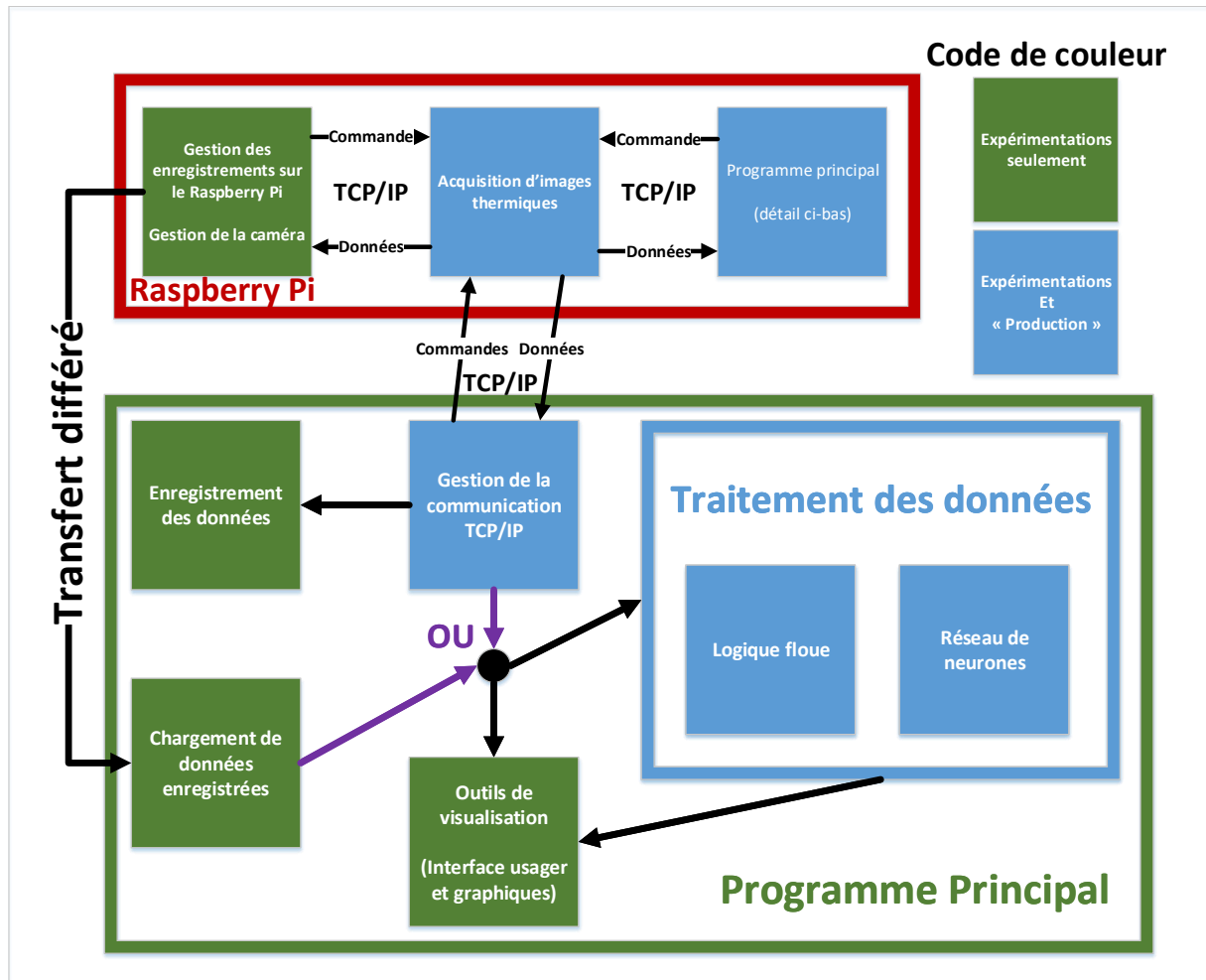


Figure 5.7 Schéma de l'architecture logicielle

Le matériel étant en place, regardons maintenant l'aspect logiciel. La figure 5.7 donne une vue globale de l'architecture logicielle développée.

La partie supérieure de la figure représente le Raspberry Pi, qui est composé de trois blocs. Un de ces blocs est le programme principal qui est détaillé juste en dessous. On remarque également que tous les éléments de cette architecture ne sont pas utilisés en même temps. En effet, les blocs en vert ne sont utilisés que pour les expérimentations alors que ceux en bleu seraient également utiles si on considère une situation d'utilisation réelle du système.

Ensuite, on remarque que l'échange de données entre le programme d'acquisition des données (qui gère l'interface avec le capteur), se fait à l'aide du protocole TCP/IP. Cette communication, dans le cadre des expérimentations, peut se faire vers un autre programme

directement sur le Raspberry Pi pour enregistrer non seulement les données de la caméra thermique, mais également les images du module caméra.

Ce sont ces enregistrements qui seront ensuite téléchargés sur un autre système pour le développement et les tests.

Il existe également la possibilité que le système distant, avec le programme principal, reçoive en direct les informations du capteur. Même si on peut alors enregistrer ces données, on comprendra que le module caméra n'étant pas impliqué, nous ne pourrions enregistrer que les données du capteur.

Finalement, on remarque la présence d'un bloc « Programme principal » sur le Raspberry Pi pour évoquer que le traitement pourrait s'effectuer en temps réel sur ce système ordonné.

Portons maintenant notre attention sur le contenu du programme principal.

On note la présence de deux méthodes par lesquelles les données peuvent être utilisées : réception par communication TCP/IP ou en chargeant un enregistrement préalable.

Une fois ces données disponibles, l'interface usager, s'il est activé, reçoit une copie de ces données pendant qu'une autre copie est transmise au traitement. Ce traitement, qui peut se réaliser à l'aide de l'une des deux approches (logique floue ou réseau de neurones), peut alors évaluer le danger et communiquer les informations pertinentes à l'outil de visualisation, si l'interface usager est activée.

IL est utile de mentionner que tout ce qui est visible à la figure 5.7 (et même plus) a été développé dans le cadre de ces travaux. Par exemple, le code nécessaire pour interfacer le capteur MLX90620 de Melexis a été réalisé en C et peut être utilisé à l'aide d'échange de message TCP/IP. Le reste du code a été développé à l'aide du langage de programmation Python qui a facilité le développement en rendant disponibles diverses bibliothèques (pour l'interface graphique, la génération des figures, le calcul matriciel, l'utilisation de logique floue, l'utilisation de JSON, de la communication TCP/IP, l'utilisation du module caméra du Raspberry Pi, l'utilisation de fichiers de configuration et finalement la parallélisation du code pour obtenir une interface réactive même pendant les traitements).

Ainsi, le code gérant l'acquisition, les communications, la gestion des enregistrements, les interfaces de visualisations, l'extraction de paramètres, l'utilisation de la logique floue et même tout le code pour l'utilisation de réseaux de neurones a été développé spécifiquement pour ce projet.

Dans les prochaines sections, nous verrons un peu plus en détail comment ces blocs fonctionnent.

5.2.1 Acquisition d'images thermiques

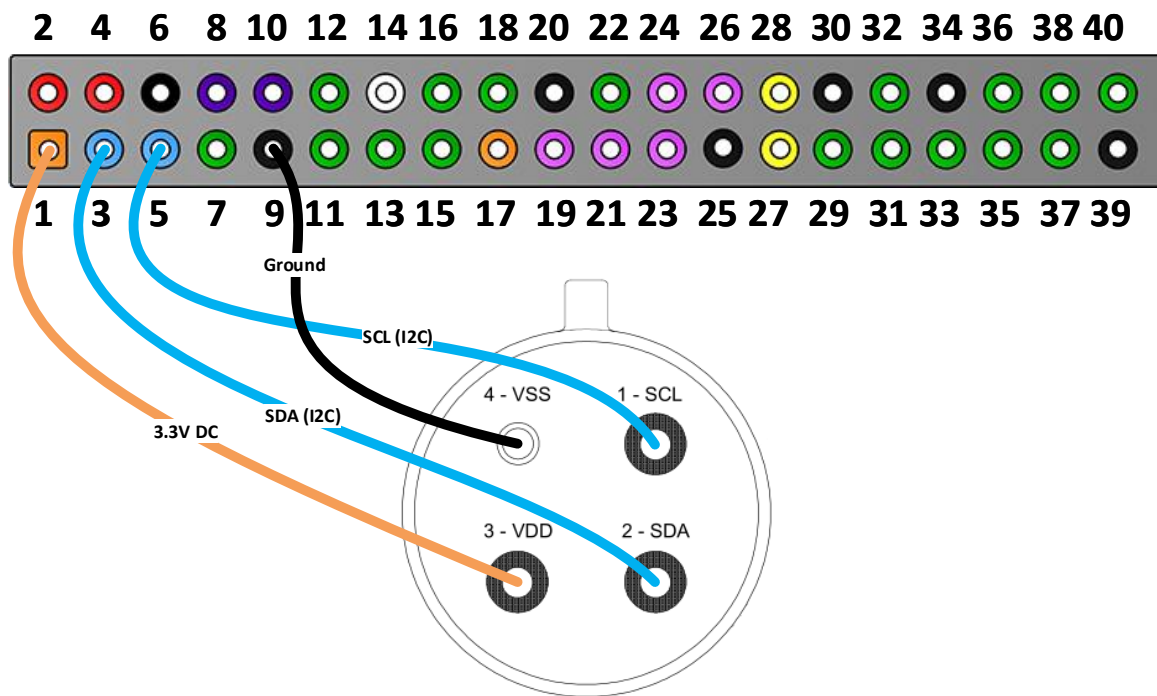


Figure 5.8 Connexions matérielles entre le capteur MLX90620⁴ de Melexis et le Raspberry Pi

La première étape consiste à permettre au système ordonné d'accéder aux données de la caméra thermique. Pour ce faire, une interface a été développée en C pour permettre de recevoir des requêtes réseau (TCP/IP) et de répondre aux diverses demandes en utilisant les données reçues du capteur. De plus, ce code implique une communication de type I_2C entre le capteur et le Raspberry Pi (figure 5.8, Annexe C pour les détails des connexions sur le Raspberry Pi). En voici les grandes lignes :

1. Réception d'une demande d'images thermique
2. Initialisation de la communication vers le capteur
3. Lecture des dernières valeurs disponibles
4. Calculs pour obtenir les températures à partir des données brutes
5. Préparation des données dans le format déterminé
6. Envoi des valeurs de température au client

⁴La partie de la figure illustrant le capteur est extraite de sa documentation officielle.

7. Répéter les étages 3 à 6 tant que l'on ne reçoit pas de requête d'arrêt de la part du demandeur

En ce qui a trait au capteur, on peut voir à la figure 5.9 l'étape d'initialisation, la lecture des valeurs mesurées suivie d'un peu plus de détails sur les étapes impliquées dans le calcul requis pour obtenir des températures.

Au sujet de ces calculs, les équations 5.1 sont un extrait de la documentation officielle du capteur (MLX90620).

$$T_{O(i,j)} = \sqrt[4]{\frac{V_{IR(i,j)_COMPENSATED}}{\alpha_{(i,j)}} + (T_a + 273.15)^4} - 273.15 \quad (5.1a)$$

$$V_{IR(i,j)_COMPENSATED} = \frac{V_{IR(i,j)_TGC_COMP}}{\varepsilon} \quad (5.1b)$$

$$V_{IR(i,j)_TGC_COMP} = V_{IR(i,j)_OFF_COMP} - \frac{TGC}{32} \cdot V_{IR_CP_OFF_COMP} \quad (5.1c)$$

$$V_{IR_CP_OFF_COMP} = V_{CP} - (A_{CP} + \frac{B_{CP}}{2^{B_{i_SCALE}}}(T_a - T_{a_0})) \quad (5.1d)$$

$$V_{IR(i,j)_OFF_COMP} = V_{IR(i,j)} - (A_{i(i,j)} + \frac{B_{i(i,j)}}{2^{B_{i_SCALE}}}(T_a - T_{a_0})) \quad (5.1e)$$

On peut y voir à l'équation 5.1a que pour déterminer la température de l'objet distant d'un pixel ($T_{O(i,j)}$, où i et j décrivent la position du pixel dans la matrice), nous avons besoin de la valeur compensée du pixel en question ($V_{IR(i,j)_COMPENSATED}$), d'un coefficient de sensibilité du pixel ($\alpha_{(i,j)}$) qui est conservé dans une mémoire interne du capteur (EEPROM) ainsi que de la température ambiante (T_a). Rappelons qu'à la section 4.1 nous mentionnions qu'une thermopile dépend de la différence de température entre ses deux jonctions (T_a et $V_{IR(i,j)}$ selon la nomenclature utilisée dans la documentation).

Les équations 5.1b à 5.1e décrivent ensuite comment $V_{IR(i,j)_COMPENSATED}$ est calculé (ce qui intègre la compensation, normalisation et compensation de l'émissivité évoquées à la figure 5.9). On y voit apparaître plusieurs autres variables. Le tableau 5.1 permet d'en connaître la signification.

En résumé, on doit d'abord déterminer la température ambiante et ensuite effectuer divers ajustements. Le premier consiste à ajuster les valeurs lues selon une compensation conservée dans une mémoire sur le capteur (déterminée par le fabricant). Ensuite, une

Variable	Définition
$\alpha_{(i,j)}$	Coefficient individuel de sensibilité du pixel (i, j) (récupéré du EEPROM du capteur)
ε	Coefficient d'émissivité (récupéré du EEPROM du capteur)
A_{CP}	Coefficient de compensation pour V_{CP} (récupéré du EEPROM du capteur)
$A_{i(i,j)}$	Coefficient de compensation pour le pixel (i, j) (récupéré du EEPROM du capteur)
B_{CP}	Pente de la relation du pixel de compensation et de T_a (récupéré du EEPROM du capteur)
$B_{i(i,j)}$	Pente de la relation du pixel (i, j) et de T_a (récupéré du EEPROM du capteur)
B_{i_SCALE}	Coefficient de mise à l'échelle de la pente de compensation (récupéré du EEPROM du capteur)
T_a	Température ambiante du capteur (jonction froide du thermocouple) (calculé à partir d'une lecture et de valeurs du EEPROM)
T_{a_0}	Constante : 25°C
TGC	Coefficient de gradient thermique (récupéré du EEPROM du capteur)
$V_{IR(i,j)}$	Lecture à la jonction chaude du thermocouple pour le pixel (i, j) , (récupéré dans la RAM du capteur)
V_{CP}	Valeur de compensation (lecture des pixels) (récupéré dans la RAM du capteur)

Tableau 5.1 Glossaire des variables utilisées dans les calculs du capteur

compensation pour l'émissivité de l'objet dont on mesure la température est appliquée aux données pour enfin pouvoir déterminer la température de l'objectif.

⁵Ce diagramme est une version traduite et simplifiée de la figure 3 de sa documentation officielle du capteur.

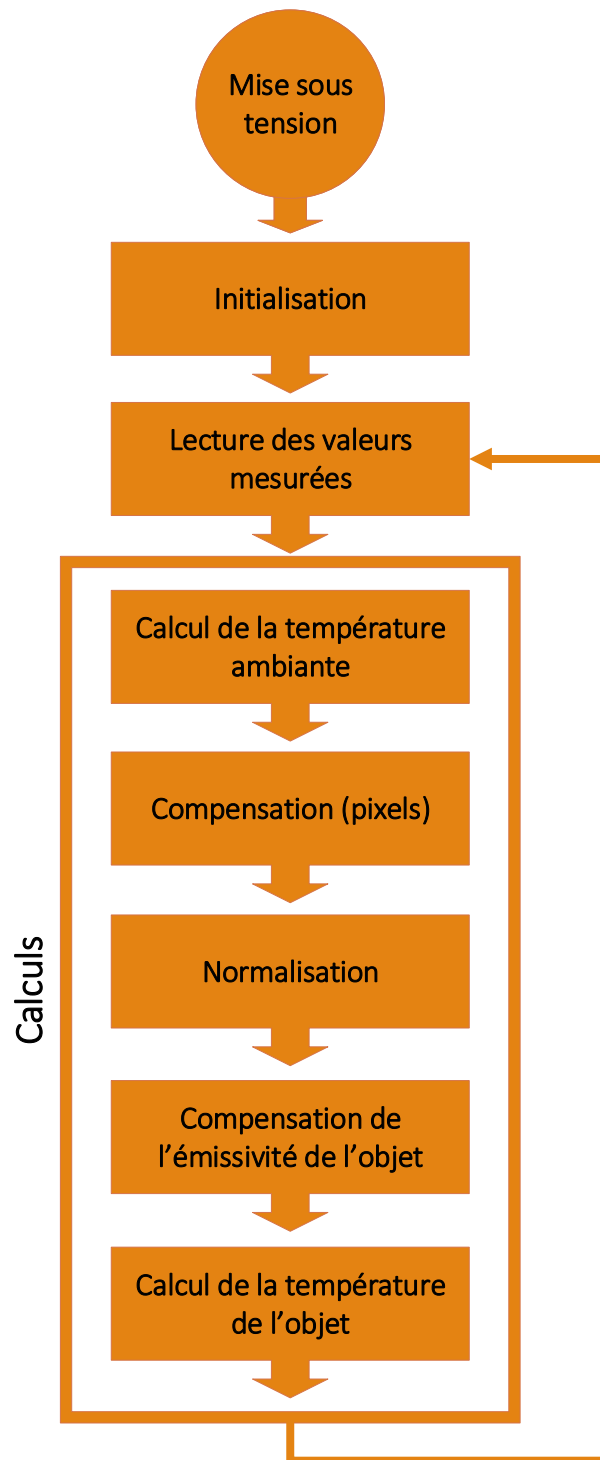


Figure 5.9 Étapes d'utilisations du capteur MLX90620 de Melexis⁵

5.2.2 Enregistrement des données

Afin de pouvoir utiliser les mêmes données à plusieurs reprises, ce qui s'avère indispensable pour les tests, les données reçues du capteur sont enregistrées. À cette fin, une partie de code que l'on pourrait qualifier de « client » demande au Raspberry Pi de lui faire parvenir les données du capteur et peut les enregistrer et les traiter en même temps. Il est également possible que ce code « client » soit utilisé directement sur le Raspberry Pi. C'est d'ailleurs ce qui a permis de réaliser un montage autonome pour réaliser les différents enregistrements mentionnés à la section 5.1.1.

Les données sont enregistrées dans un seul fichier qui comporte les paramètres de l'enregistrement en tant que première ligne et ensuite une image par ligne. Le tout étant encodé en JSON[JSO, 2017] pour en faciliter l'accès si l'on désire réutiliser les données à partir d'un autre programme en plus de permettre de consulter directement le fichier (les données sont lisibles en tant que texte, contrairement à un fichier binaire qui demande un programme de lecture spécifique). Dans le cas particulier des enregistrements ayant servi aux tests et au développement, ce fichier d'images thermique était accompagné d'un dossier contenant un fichier image pour chaque image thermique. Ces images provenaient du module caméra du Raspberry Pi et permettaient de faciliter l'utilisation des données pendant les tests (on pouvait voir à la fois l'image thermique et la scène originale, ce qui facilite l'interprétation des données).

La figure 5.10 donne un aperçu de l'interface développée qui permettait de voir à la fois les 2 images. On y voit 4 rangées par 16 colonnes de « pixels » de température où la couleur passe du bleu au rouge plus la température est élevée. On peut également y voir la température de ce point de mesure en blanc. Sous cette grille, on peut voir la température ambiante détectée par le capteur (dénnoté « TA »), l'heure associée à l'image thermique, le minimum et maximum de température rapporté par le code interface et la fréquence d'échantillonnage. Plus bas, l'image du module de caméra est affichée en incluant la date et l'heure de capture de cette image en haut de celle-ci. Il est donc possible de valider que les images correspondent au même moment.

Par la suite, il est possible de réutiliser un enregistrement et l'interface affiche alors des contrôles simples pour avancer, reculer, sauter à une image ou simplement mettre en pause ou reprendre la lecture (tel que l'on peut le voir à la figure 5.11).

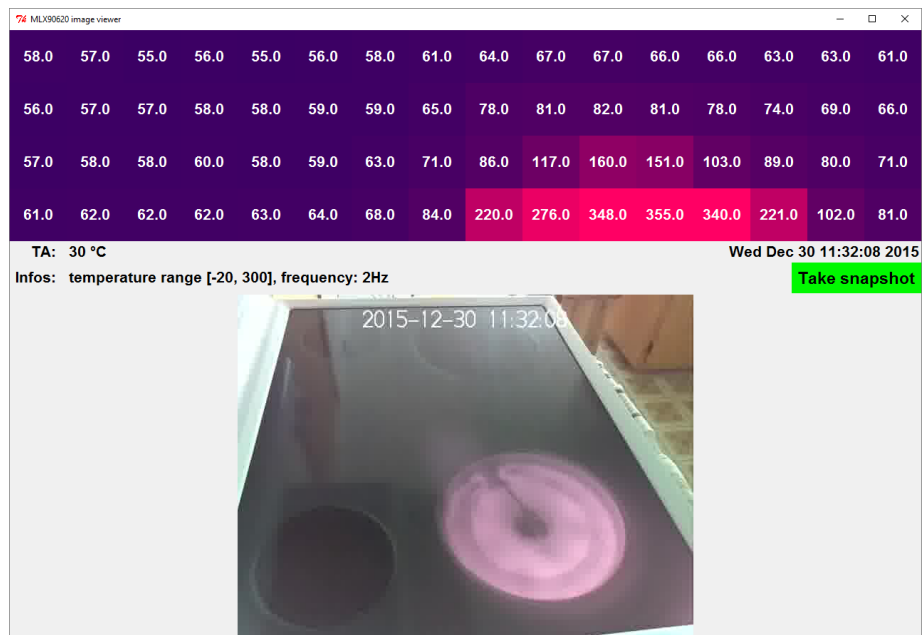


Figure 5.10 Interface de visualisation



Figure 5.11 Interface permettant de rejouer un enregistrement préalable

5.2.3 Scénarios d'expérimentation

De façon générale, on peut décrire les enregistrements ayant servi aux expérimentations comme étant une combinaison des situations suivantes :

1. Surface de cuisson vide (surtout au début d'un enregistrement)
2. Rond allumé à puissance maximale non recouvert (température augmentant rapidement)
3. Main se déplaçant à proximité de la source de chaleur (mitaine de four lorsqu'il y a un réel danger de brûlure)
4. Ustensile de cuisson (casserole) rempli d'eau, l'eau étant graduellement portée à ébullition
5. Déplacement de l'ustensile de cuisson sur la cuisinière (peut représenter le début ou la fin d'une cuisson)

Identification	Situations	Durée	Utilisé pour l'approche
TestAI	1, 2, 3, 4, 5	1385 images (11m32.5s)	Logique floue et réseau de neurones
TestAI1	1, 2, 3, 4, 5	140 images (70s)	Logique floue et réseau de neurones
TestAI1_subset	1, 2, 3, 4	18 images	Réseau de neurones
TestAI2	2, 3, 4, 5	129 images (64.5s)	Logique floue et réseau de neurones

Tableau 5.2 Liste des enregistrements utilisés pour le développement

Dans un des enregistrements les plus utilisés (portant l'appellation « TestAI1 » à l'interne), toutes les situations mentionnées sont présentes dans un enregistrement de 140 images, donc de 70 secondes. Bien entendu, le délai est trop court pour porter l'eau à ébullition, mais on dépose tout de même une casserole contenant de l'eau sur le rond vers la fin de l'enregistrement. Comme cet ustensile demeure en place jusqu'à la fin des 140 images, nous n'avons en fait un déplacement de l'ustensile qu'en direction du rond.

« **TestAI2** » vient compléter « TestAI1 » en combinant l'eau portée à ébullition dès le début de l'enregistrement. Il y a alors dégagement de vapeur d'eau et une mitaine de four qui passe entre la casserole et la caméra thermique. Aussi, la casserole est déplacée sur la surface de cuisson, laissant le rond allumé sans ustensile.

Vient ensuite un autre enregistrement, « TestAI », plus long cette fois (1385 images), qui permet de voir l'eau passer de la température de la pièce jusqu'à l'ébullition. On y voit

également la casserole être déplacée, une mitaine de four passer au-dessus de la source de chaleur et un rond exposé

Un sous-échantillonnage a également été réalisé sur « TestAI1 » pour en faire un nouveau scénario pour l'entraînement du réseau de neurones. Il s'agit de 18 images récupérées à différents endroits dans « TestAI1 ».

D'autres enregistrements ont été réalisés, de situations réelles de cuisson, mais n'ont pas été utilisées pour le développement. Ces derniers étant généralement trop long (souvent plus de 30 minutes) et ne comportant pas assez de situations à risque pour permettre un entraînement ou une validation de performance des méthodes de traitement. Cependant, deux exceptions à cette règle existent, tel que décrit dans le tableau 5.3 qui décrit également les enregistrements mentionnés précédemment au tableau 5.2.

Identification	Description
TestAI	Enregistré exclusivement pour les expérimentations ; pour avoir des données sur une casserole d'eau portée à ébullition. On profite de l'occasion pour avoir un rond non recouvert à puissance maximale, une main à proximité du rond chaud découvert (simulation de risque de brûlure) ainsi qu'une mitaine de four au-dessus de l'eau à ébullition (simulation de risque de brûlure).
TestAI1	Enregistré exclusivement pour les expérimentations ; pour condenser en un temps relativement court les situations d'un rond allumé à puissance maximale sans ustensile, le passage d'une main au-dessus de ce rond chaud (simuler un risque de brûlure) et finalement une casserole remplie d'eau qui est déposée sur le rond (rendant la situation normale, donc sécuritaire)
TestAI1_subset	Sous-ensemble des données de TestAI1 où l'on retrouve six groupes de 3 images consécutives extraites en vue de l'entraînement d'un réseau de neurones
TestAI2	Enregistré exclusivement pour les expérimentations ; pour avoir, en un temps restreint, une casserole dont l'eau est déjà à ébullition, une mitaine se déplaçant au-dessus de la casserole (dans la vapeur chaude, simulant un risque de brûlure), un rond allumé non recouvert et le retour de la casserole d'eau bouillante sur le rond
cuisine2	Un enregistrement dans une 2e cuisine, reprenant sensiblement le contenu de « TestAI1 ». Cet enregistrement n'a été utilisé que pour produire un comparatif dans un environnement similaire, mais différent (valider l'adaptabilité du système). 163 images (81.5s)
2016-01-03_18-46-38	Enregistrement d'une situation réelle de cuisson (préparation de spaghetti). Il s'agit ici aussi de produire une comparaison des résultats en dehors d'une situation de tests. 3085 images (25m42.5s)

Tableau 5.3 Description des enregistrements utilisés

5.2.4 Traitement des données

Logique floue

Le traitement à l'aide de la logique floue (tel que décrite à la section 4.3.1) a débutée par explorer différents paramètres extraits.

Comme il s'agit d'utiliser des règles pour représenter le raisonnement d'un expert, les expérimentations ont débuté par des essais simples tel que de considérer qu'une température plus élevée implique une augmentation du risque.

Cependant, on atteint assez rapidement des limites de précision, surtout pour détecter la présence d'une main (par exemple, pour déterminer s'il y a risque de brûlure).

Une exploration plus formelle des tous les paramètres extraits à l'aide de graphiques et de tableaux pour tenter de détecter un patron lors des moments évalués manuellement à risque a donc été nécessaire.

Les règles qui en résultaient étaient ensuite mises à l'épreuve en les utilisant sur des données de tests (principalement **TestAI1**, mais également **TestAI** et **TestAI2**) pour en évaluer la performance.

Rappelons qu'à la section 4.3.1 ont été présentées des figures (4.7, 4.8, 4.9 et 4.10) démontrant la possibilité de détecter une occlusion (que l'on peut supposer liée à la proximité de l'utilisateur).

Essentiellement, les règles utilisées considèrent deux aspects. Le premier étant la variation de la température minimale de l'image complète en sens inverse du minimum d'une zone contigüe (« blob ») de chaleur élevée pour détecter une occlusion. Le deuxième, utilise la logique simple du début : si la température augmente, le risque de la situation augmente également. Ainsi, la température maximale moyenne des dernières images (nombre variable selon la configuration) est utilisée en combinaison avec la détection d'occlusion.

Réseau de neurones

L'approche à base d'un réseau de neurones (tel que décrite à la section 4.3.2) a débuté assez simplement, avec un code ne permettant qu'une seule couche cachée et sans aucuns biais.

Ce code a par la suite été étendu pour permettre une plus grande souplesse. En effet, pour faciliter l'exploration de différentes configurations du réseau et donc obtenir de bonne

performance, il était plus pratique d'avoir un code unique permettant la création de réseaux de différentes formes.

Une généralisation des étapes s'en est donc suivi pour en arriver à l'équation 4.23. Ainsi, il était possible de créer un réseau selon une configuration des couches cachées à l'aide d'une liste passée en paramètre. Cette liste comportait, pour chaque couche, le nombre de neurones à utiliser.

Un autre ajout fut l'intégration d'un biais. Pour ce faire, une entrée fictive pour chaque couche, excepté la couche d'entrée, était ajoutée. Cette entrée était ensuite reliée à tous les neurones de la couche en question, tout comme pour les autres entrées (comme on peut le voir à la figure 4.15).

Le rôle de ce biais est de permettre de déplacer latéralement notre fonction sigmoïdale 4.13 tel que l'on peut le constater à la figure 5.12. Aussi, bien que la valeur de ce biais soit fixée à 1, les poids associés aux synapses le reliant aux neurones pouvant être modifiés pendant l'entraînement, son impact pourra être modulé. Ainsi, sa contribution pourra être autant positive que négative, autant grande que petite.

Enfin, une dernière amélioration a été apportée à la paramétrisation de ce code pour permettre l'utilisation de plusieurs images à la fois. En effet, comme nous avons tenté d'utiliser des caractéristiques temporelles avec l'approche utilisant la logique floue, il était intéressant de répéter l'expérience avec un réseau de neurones.

Un nombre variable d'images combinées en tant qu'une seule entrée a donc été prévue pour que le réseau « voie » plusieurs images simultanément. Ainsi, si on utilise une configuration avec 3 images, au lieu des 65 entrées prévues pour le réseau de neurones, nous aurons 3x 65 entrées, donc 195. Ces entrées seront les 65 informations de la première image, suivi des 65 de la deuxième et enfin des 65 de la troisième. Ensuite, les prochaines valeurs envoyées au réseau de neurones seront les informations des images 2, 3 et 4.

La seule exception à cette méthode de combinaison des images est lorsqu'il n'y a pas suffisamment d'images précédentes (au début de l'utilisation du réseau). La première image sera alors répétée le nombre de fois nécessaires pour obtenir la bonne quantité d'entrées.

La dernière étape était d'entraîner le réseau. Pour ce faire, des tests ont été réalisés avec différents enregistrements (détaillés au tableau 5.2).

Il y a le cas particulier de l'entraînement avec un sous-ensemble de données qui se distingue des autres. En effet, ces données n'ont pas servi à l'évaluation, mais plutôt uniquement à

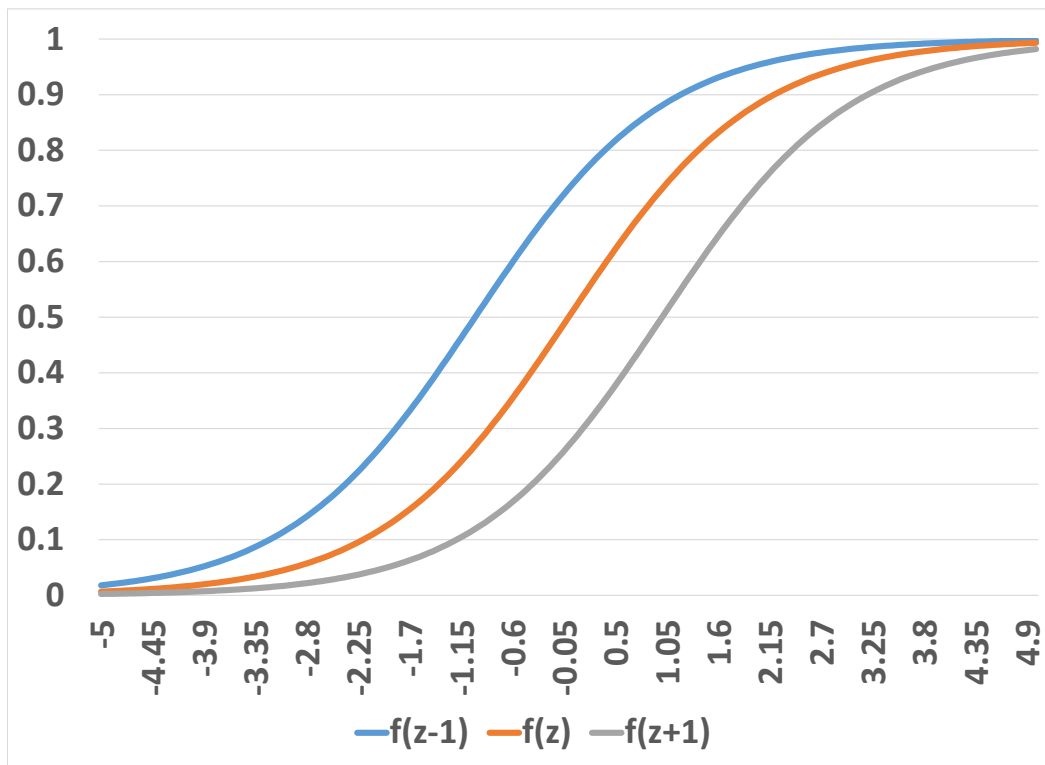


Figure 5.12 Effet du biais sur la fonction sigmoïdale

entraîner un réseau. Le but recherché était d'avoir des données non ambiguës et identifiées de façon binaire comme étant sécuritaires ou dangereuses pour entraîner le réseau.

Pour les autres ensembles de données ayant été utilisées pour l'entraînement, l'enregistrement complet était utilisé et évalué une fois de plus de façon binaire.

L'entraînement faisait appel à une fonction d'optimisation (« `scipy.optimize.minimize` ») de la librairie « Scipy » en python. Son rôle étant simplement de minimiser une fonction de coût (équation 4.8).

Le nombre d'itérations est variable, mais affiché à la fin du processus. De plus, il est possible de suivre « l'évolution » de l'entraînement pour ensuite pouvoir tracer un graphique du coût en fonction du nombre d'itérations. Il est également possible, avec les outils en place, de valider s'il y a surentraînement (« overfitting ») à l'aide d'un graphique. Cette validation nécessite de réserver une partie des données du scénario utilisé pour l'entraîne-

ment pour tester s'il y a surentrainement. Dans le cas présent, deux images successives sur trois servent à l'entraînement alors que la suivante est utilisée pour tester.

CHAPITRE 6

RÉSULTATS

Dans ce chapitre, le résultat des expérimentations, les réussites et les limitations des méthodes proposées seront abordées.

6.1 Logique floue

Tout d'abord, la première approche à avoir été mise en oeuvre fut l'utilisation de logique floue. Cette approche ayant été développée à base de règles, le but était de faire une preuve de concept que le système pouvait différencier une situation dangereuse d'une sécuritaire.

Cette preuve de concept est réussie, comme on peut le constater aux figures 6.1, 6.2, 6.3 et 6.4. Sur ces quatre figures, on retrouve une évaluation manuelle (évaluation subjective par le développeur) du danger en bleu, une évaluation calculée à partir des mêmes paramètres utilisés par la logique floue (équations 6.2, utilisant les abréviations décrites en 6.1) en orange et le résultat de l'évaluation avec l'approche à base de logique floue en mauve.

Les deux dernières courbes (orange et mauve) utilisent les mêmes données, mais de façon différente. La première (calculée, en orange) affiche une plus grande précision puisqu'il s'agit d'un calcul linéaire. En comparaison, la courbe de résultat de la logique floue affiche moins de variations. Il s'agit en fait du résultat des règles utilisées pour cette approche qui se comportent comme des paliers.

On constate donc que l'évaluation manuelle est binaire (0 étant sécuritaire et 1 une situation dangereuse), que la courbe orange (calculée) peut afficher toutes les valeurs entre 0 et 1 inclusivement alors que la logique floue se limite à quelques valeurs de sorties, toujours entre 0 et 1.

$$\text{Abréviation} = \text{nom complet du paramètre, tel que décrit à l'annexe B} \quad (6.1a)$$

$$\delta_{blob_min} = \text{delta_blobsGlobalMin} \quad (6.1b)$$

$$\delta_{min} = \text{delta_min} \quad (6.1c)$$

$$\text{max} = \text{filtered_max} \quad (6.1d)$$

$$danger = obstruction + temperature \quad (6.2a)$$

$$obstruction = \begin{cases} \frac{|\delta_{blob_min} - \delta_{min}|}{2.5}, & \text{Si } (\delta_{blob_min} < -0.5 \wedge \delta_{min} \geq 0) \vee (\delta_{blob_min} > 0.5 \wedge \delta_{min} < -0.5) \\ 0, & \text{Sinon} \end{cases} \quad (6.2b)$$

$$temperature = \begin{cases} \frac{max-45}{70}, & \text{Si } max \geq 45 \\ 0, & \text{Sinon} \end{cases} \quad (6.2c)$$

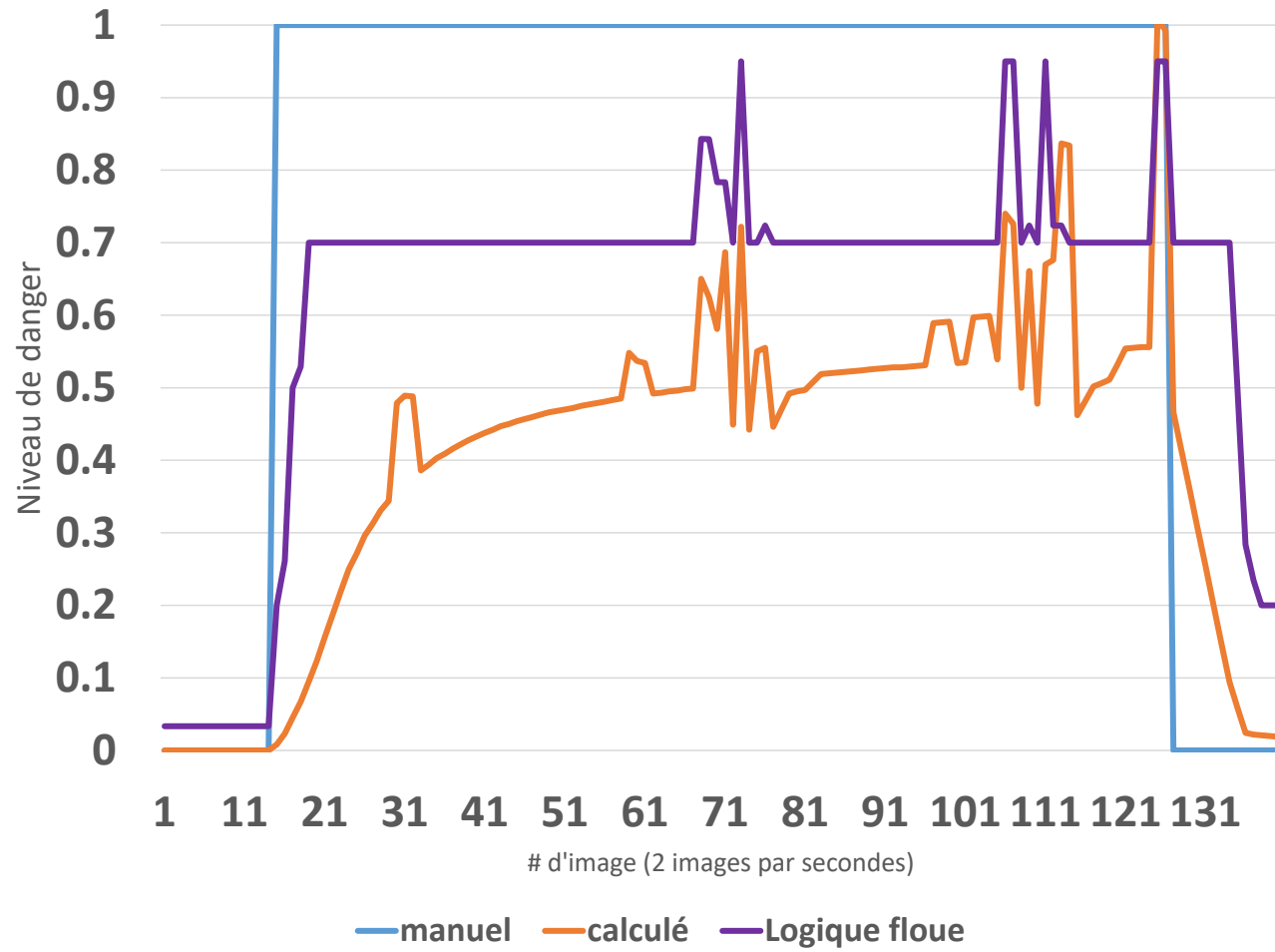


Figure 6.1 Résultat de l'utilisation de logique floue sur « TestAI1 »

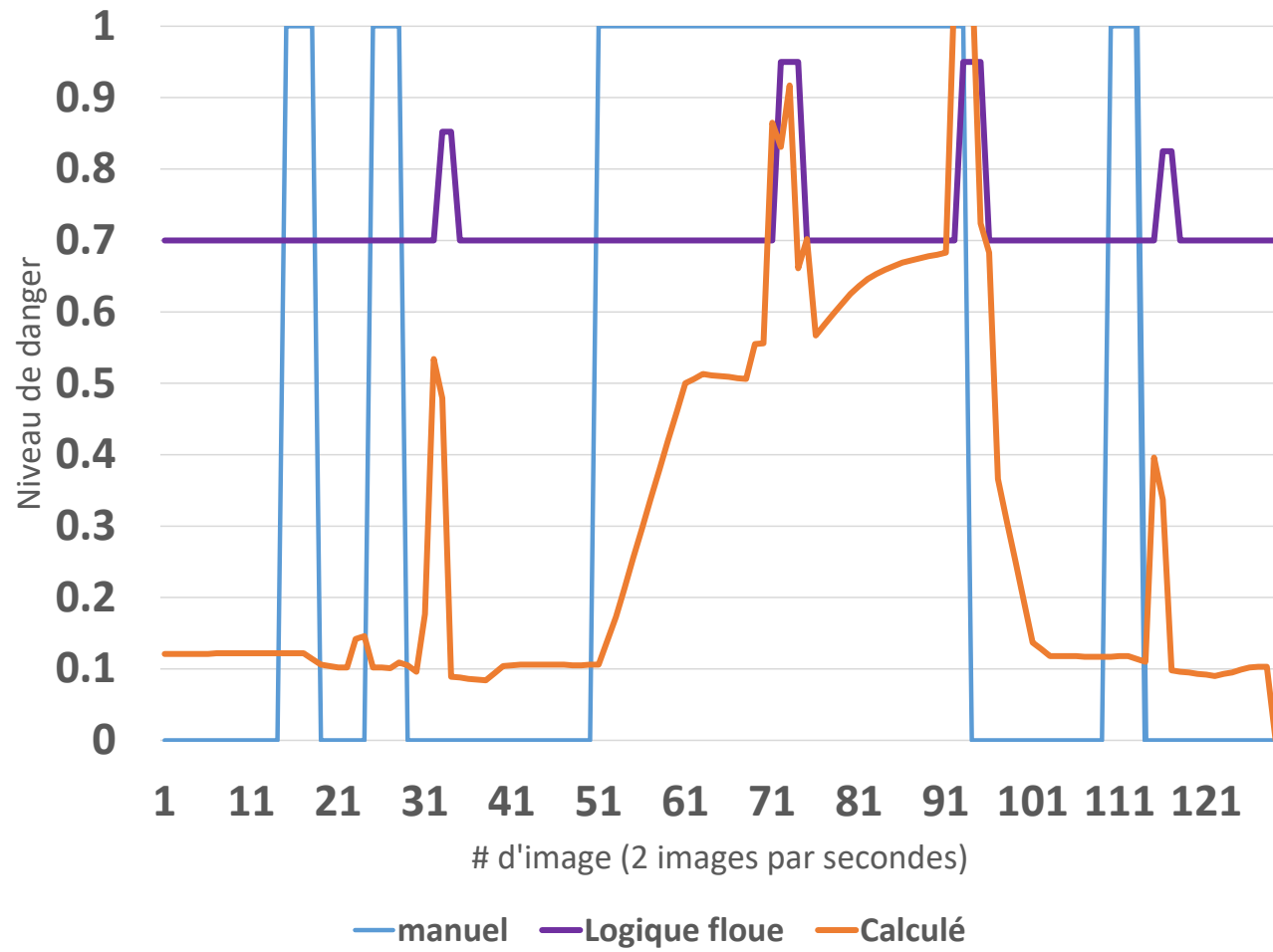


Figure 6.2 Résultat de l'utilisation de logique floue sur « TestAI2 »

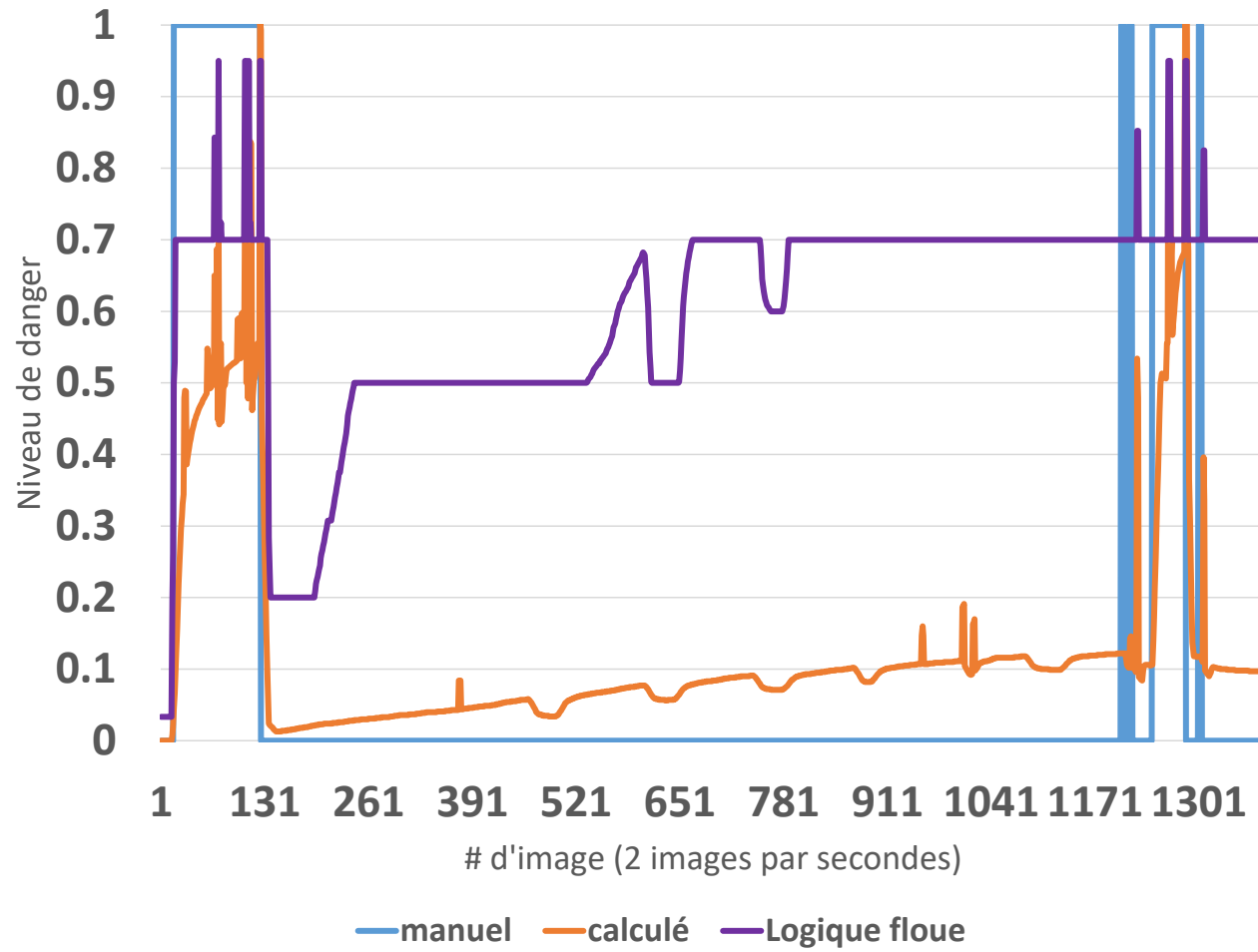


Figure 6.3 Résultat de l'utilisation de logique floue sur « TestAI »

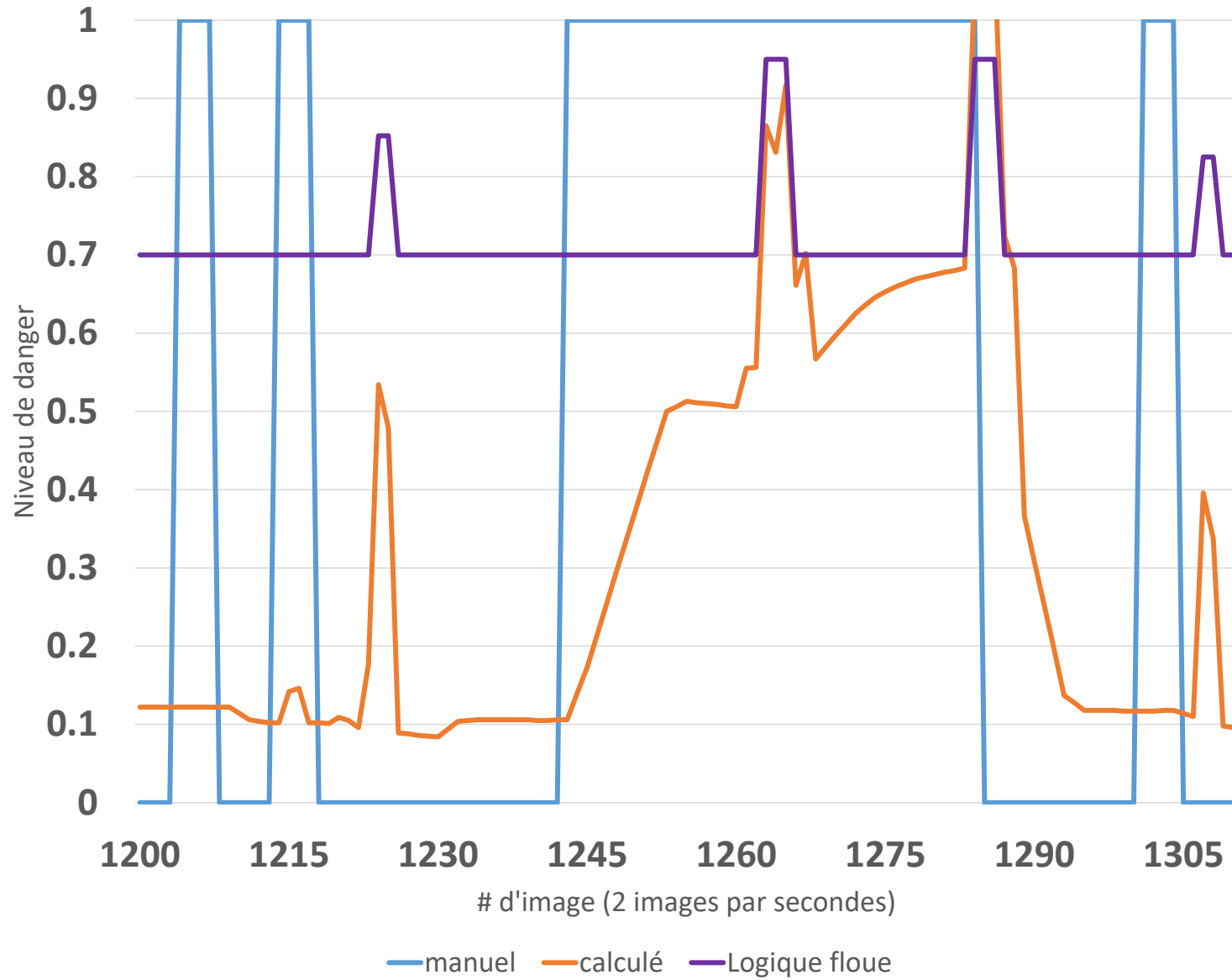


Figure 6.4 Agrandissement de la fin de la figure (des images 1200 à 1300) [6.3](#)

On constate donc que le résultat du traitement avec la logique floue détermine un niveau de danger non binaire (contrairement à l'évaluation manuelle), mais pas aussi granulaire qu'un calcul. Il serait par contre possible de rendre la sortie plus lisse et plus près de la version calculée en ajustant les règles et caractéristiques des fonctions d'appartenance utilisées.

Aussi, il est utile de savoir que les règles utilisées ont été conçues pour attribuer un danger de 0.7 à un rond exposé (température élevée et risque de brûlure s'il y a contact) et un danger s'approchant de 1 lorsqu'il y a une occlusion de détectée (proximité de l'utilisateur, donc risque plus important que pour une température élevée).

Ainsi, le travail qui devrait être fait avant d'utiliser cette approche serait de consulter plusieurs spécialistes pour déterminer les règles et paramètres des fonctions d'appartenance appropriés. Rappelons qu'un système à base de logique floue reproduit le raisonnement d'un expert. Un raffinement des règles utilisé serait donc le bienvenu.

6.2 Réseau de neurones

Pour l'approche utilisant un réseau de neurones, deux approches se sont confrontées : obtenir un résultat binaire ou qui permet d'obtenir divers niveaux de dangers.

Dans le premier cas, l'entraînement a été effectué sur un enregistrement complet avec une évaluation manuelle binaire. Les données utilisées pour cet apprentissage sont celles de « TestAI » avec une configuration de réseau identique à celle présentée à la section 4.3.2. Sur les prochaines figures, les résultats de ce réseau seront identifiés « évalué, training testAI 10_10_10-0-1 ».

Dans le deuxième cas, pour obtenir plus de variation dans le niveau de danger identifié par le système, l'apprentissage a eu lieu sur un sous ensemble de « TestAI1 » (« TestAI1_subset », figure 6.5), toujours en utilisant des valeurs binaires. La différence est ici que les images présentées ne sont aucunement ambiguës. Ainsi, le système apprend ce qui doit être détecté comme un danger maximal ou une absence totale de danger et infère les niveaux intermédiaires. Sur les prochaines figures, les résultats générés par l'utilisation de ce réseau seront identifiés « évalué, training testAI1_subset 10_10_10-0-1 ».

Ainsi, à la figure 6.6, on remarque que les deux réseaux performant sensiblement de la même façon, cet enregistrement étant pratiquement toujours « dangereux » (rond exposé sur presque toute la durée).

Dans le cas de « TestAI2 » (figure 6.7), comme cet enregistrement est « plus sécuritaire » (casserole avec eau chauffée/à ébullition), on note plus de variation entre les résultats des deux réseaux. Il devient alors plus facile de distinguer des évènements « normaux » liés à l'utilisation de la cuisinière d'une utilisation dangereuse. La courbe en noir illustre bien qu'un certain danger est présent, mais « plutôt normal » (manipulation de la casserole avec de l'eau chaude et dégagement de vapeur).

Enfin, la figure 6.8 permet non seulement de confirmer le comportement constaté avec la figure 6.7 (gradation du danger pour la courbe en noir), mais également de voir une augmentation graduelle du niveau de danger détecté entre les images 131 et 1171. Pour bien comprendre l'évolution de cette courbe, il faut rappeler que dans cet enregistrement, on porte de l'eau à ébullition. La casserole contenant de l'eau froide est déposée sur le rond à l'image 125, le niveau de danger augmente donc en même temps que la température de l'eau.

Un autre élément peut être visualisé à l'aide de cette courbe en noir : les moments d'activité du rond. On remarque en effet que celui-ci alterne entre la puissance maximale et minimale.

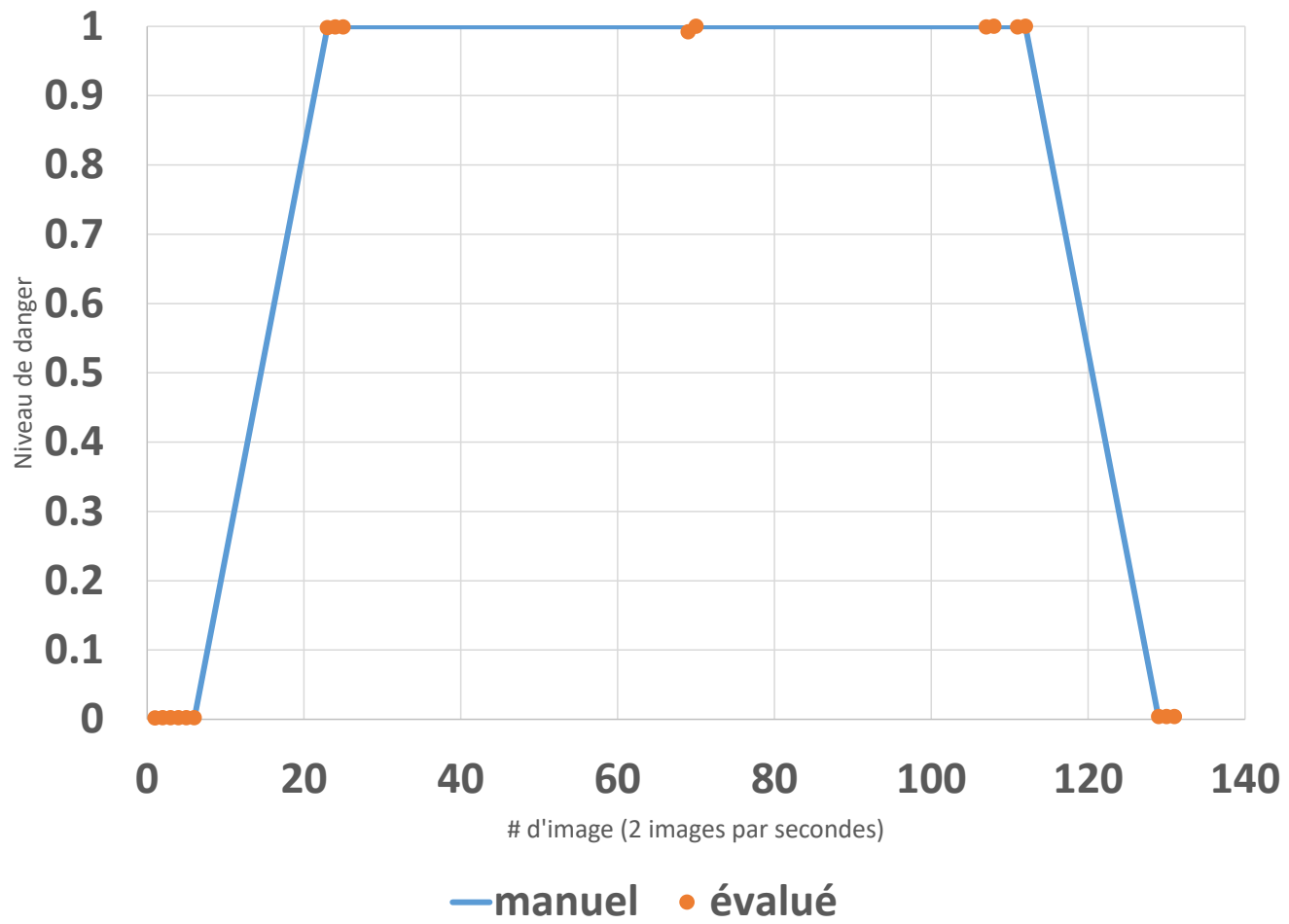


Figure 6.5 Résultat de l'entrainement d'un réseau de neurones avec « TestAI1_subset »

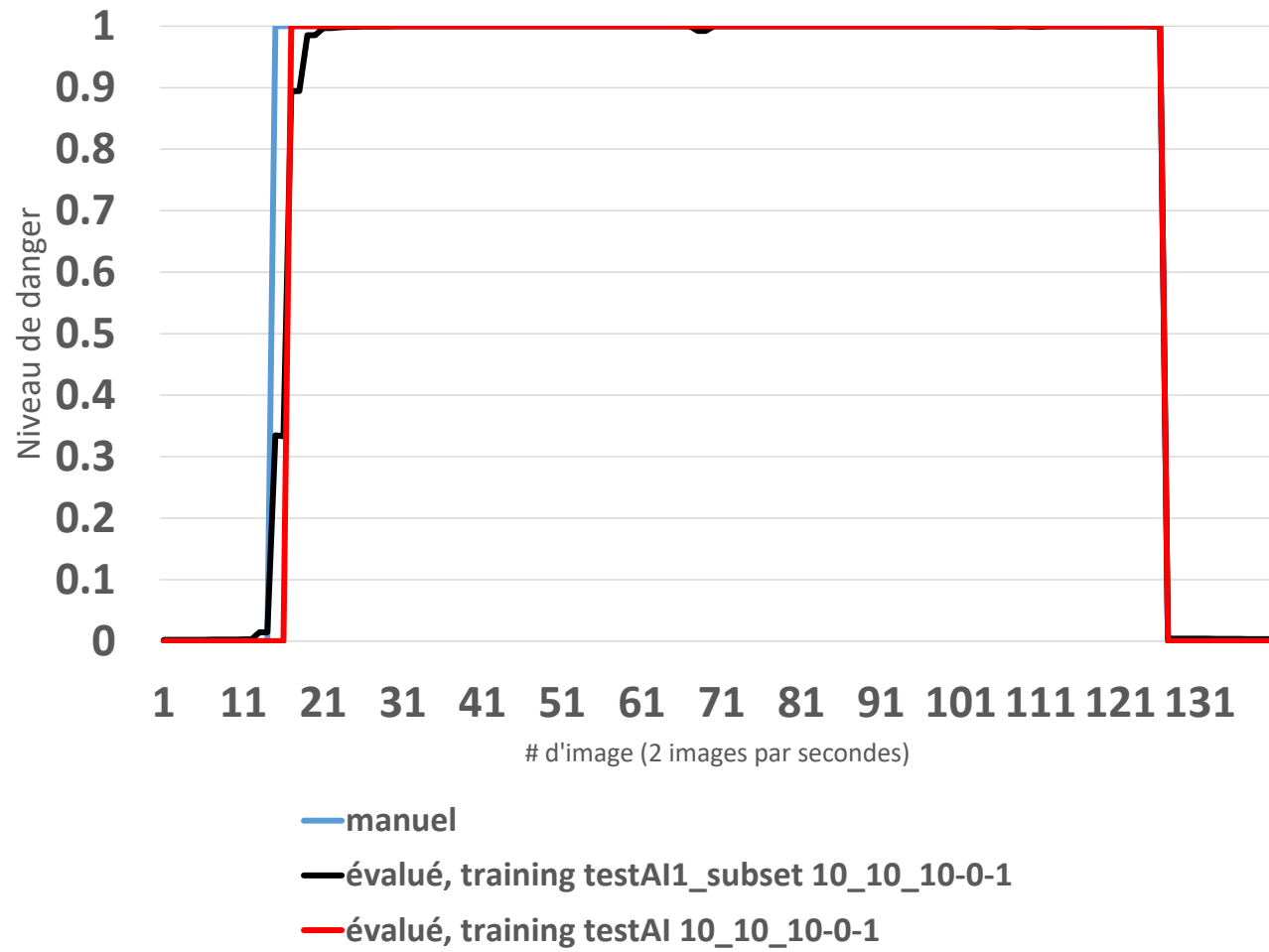


Figure 6.6 Résultat de l'utilisation de réseaux de neurones sur « TestAI1 »

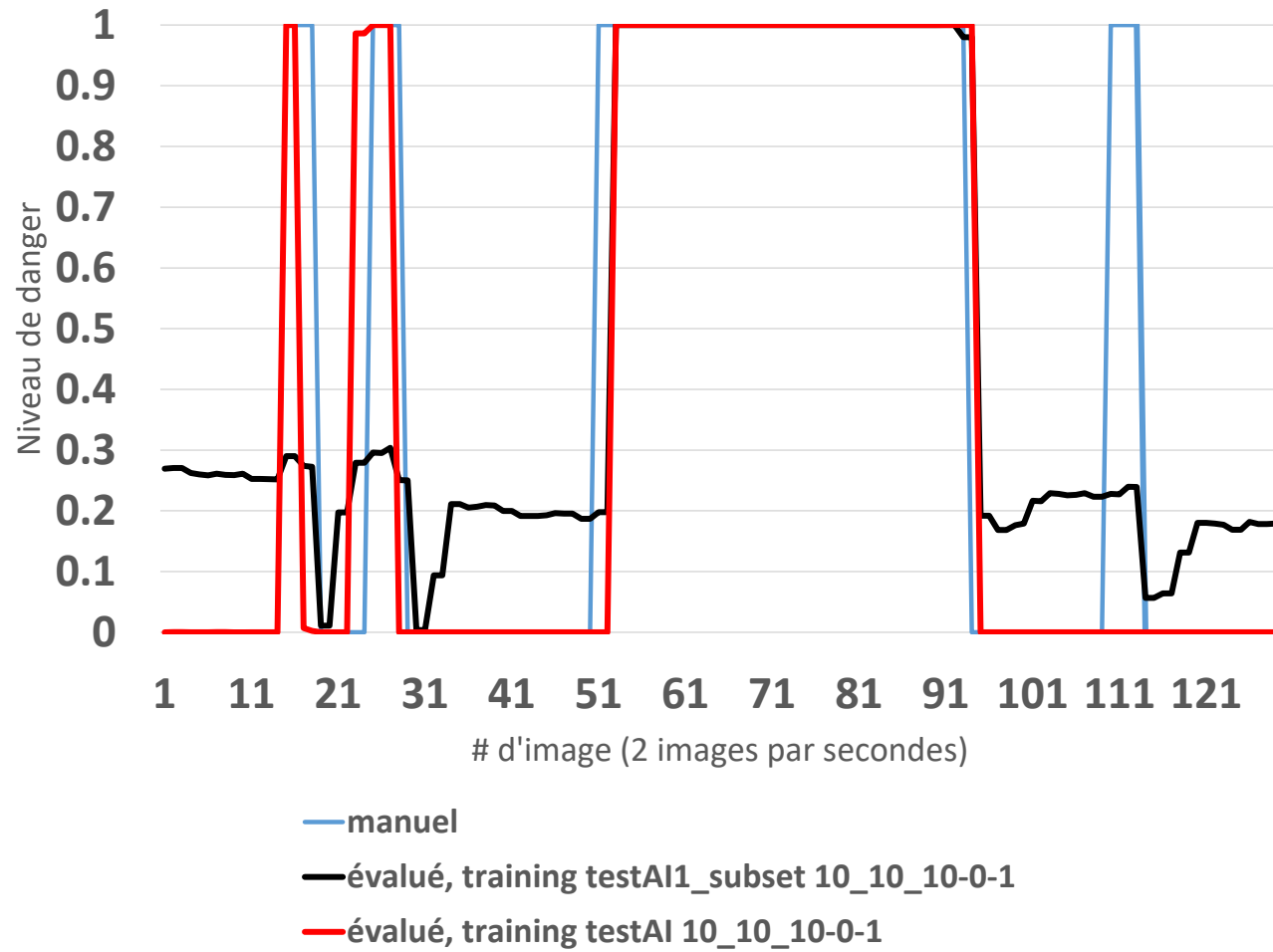


Figure 6.7 Résultat de l'utilisation de réseaux de neurones sur « TestAI2 »

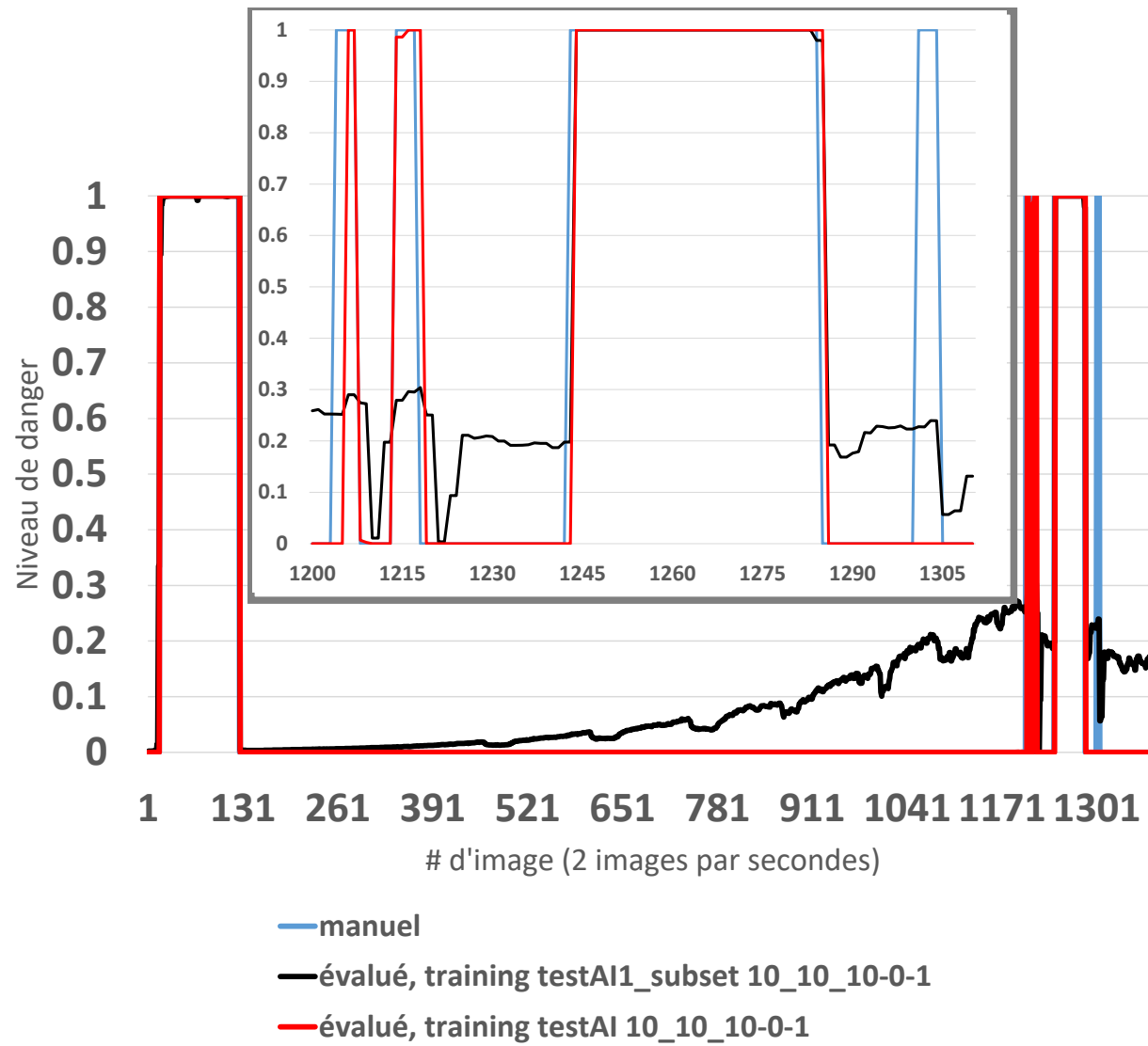


Figure 6.8 Résultat de l'utilisation de réseaux de neurones sur « TestAI »

6.3 Adaptabilité

Il est intéressant de voir comment les approches explorées se comportent lorsqu'utilisées sur des données issues d'un autre contexte.

À cet effet, deux autres enregistrements ont été utilisés : « cuisine2 » et « 2016-01-03_18-46-38 ».

Dans le premier cas, il s'agit d'un enregistrement similaire à « TestAI1 », mais dans une autre cuisine. La figure 6.9 permet de voir en un seul coup d'oeil les résultats des deux approches.

On peut y voir que les deux courbes des réseaux de neurones sont semblables (la verte, comme pour les autres enregistrements, donne un résultat binaire). L'approche à l'aide de logique floue quant à elle permet de différencier les sources de danger.

En effet, comme les règles ont été créées pour attribuer un niveau de danger plus élevé à la présence de l'utilisateur à proximité d'une source de chaleur élevée, les moments où le danger passe au-dessus de 0.7 représentent la présence de l'utilisateur.

Il existe cependant des limitations à cette détection. Par exemple, un peu avant l'image 40, l'utilisateur se trouve à côté de la cuisinière (tiroir adjacent à la cuisinière, du côté opposé au rond utilisé.). Bien qu'il ne s'agisse pas d'un faux positif, le niveau de danger ne devrait probablement pas être équivalent à une main juste au-dessus du rond allumé.

Aussi, une main passe près du rond un peu avant que la casserole ne soit apposée sur le rond chaud. Ce passage n'est pas détecté vers les images 60.

On comprendra que la détection de l'utilisateur se base sur la détection d'occlusion entre la caméra thermique et la source de chaleur. Ainsi, si la main de l'utilisateur n'est pas « visible » pour le capteur, même s'il y a proximité, le système ne peut en tenir compte.

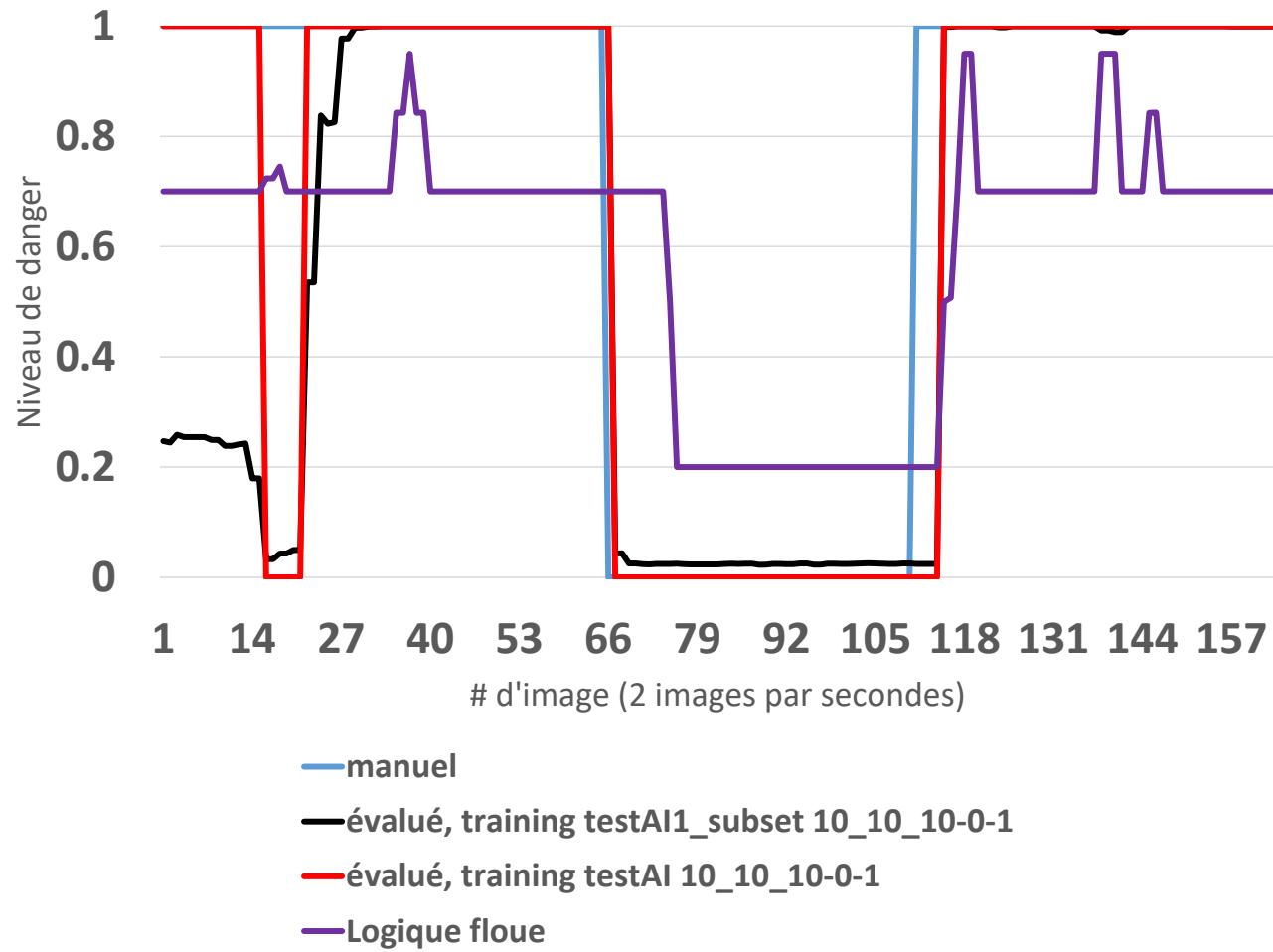


Figure 6.9 Résultat de l'utilisation des deux approches sur « cuisine2 »

Dans le deuxième cas, les approches ont été appliquées sur un enregistrement beaucoup plus long et plus près de la réalité d'utilisation de la surface de cuisson. On parle ici de la figure 6.10 pour l'enregistrement « 2016-01-03_18-46-38 ».

Cet enregistrement est en fait la préparation de spaghetti dans une casserole d'eau porté à ébullition.

On y remarque d'abord l'absence d'évaluation manuelle (de par la durée de l'enregistrement et la difficulté à juger ce qui devrait être considéré dangereux). On peut cependant considérer l'enregistrement comme « sécuritaire » avec un niveau de danger lié à la température pour porter l'eau à ébullition et aux interactions avec l'ustensile pour y placer les pâtes et brasser par exemple. On pourrait donc qualifier ce danger de « normal » ou « habituel » dans les circonstances.

On trouve donc sur la figure 6.10 les courbes associées aux deux réseaux de neurones ainsi que la courbe de l'approche avec logique floue et la version calculée à partir des mêmes paramètres à titre de référence.

Au niveau de la logique floue, on remarque qu'on ne semble pas détecter d'occlusion, le niveau reste donc à 0.7 sur une grande partie de l'enregistrement à cause de la température maximale détectée.

Le résultat est similaire avec les réseaux de neurones. En effet, le résultat binaire de la courbe en vert correspond à celui indiqué par la logique floue. Une gradation un peu plus fine est disponible avec la courbe noire. On y voit une fois de plus les moments où le rond dégage de la chaleur et l'augmentation de température de l'eau et de l'ustensile.

Une fois de plus, quelques anomalies sont présentes, principalement sur la courbe en vert. Par exemple, les deux premières barres verticales (autour de 511) ne semblent pas explicables (probablement lié à l'éclairage) alors que le pic descendant autour de l'image 2041 arrive au moment où l'utilisateur agit les pâtes. On devrait donc avoir un niveau de danger plus élevé considérant la proximité.

Ces résultats sont cependant la preuve que les deux approches développées s'adaptent bien à des situations diverses et proches de la réalité. Il reste cependant du travail à faire pour déterminer les niveaux de danger objectif selon la situation liée à l'utilisation de la cuisinière.

En effet, il serait intéressant d'abaisser le niveau détecté pour les activités de cuisson jugées « normales » et ainsi donner une rétroaction plus juste à l'utilisateur.

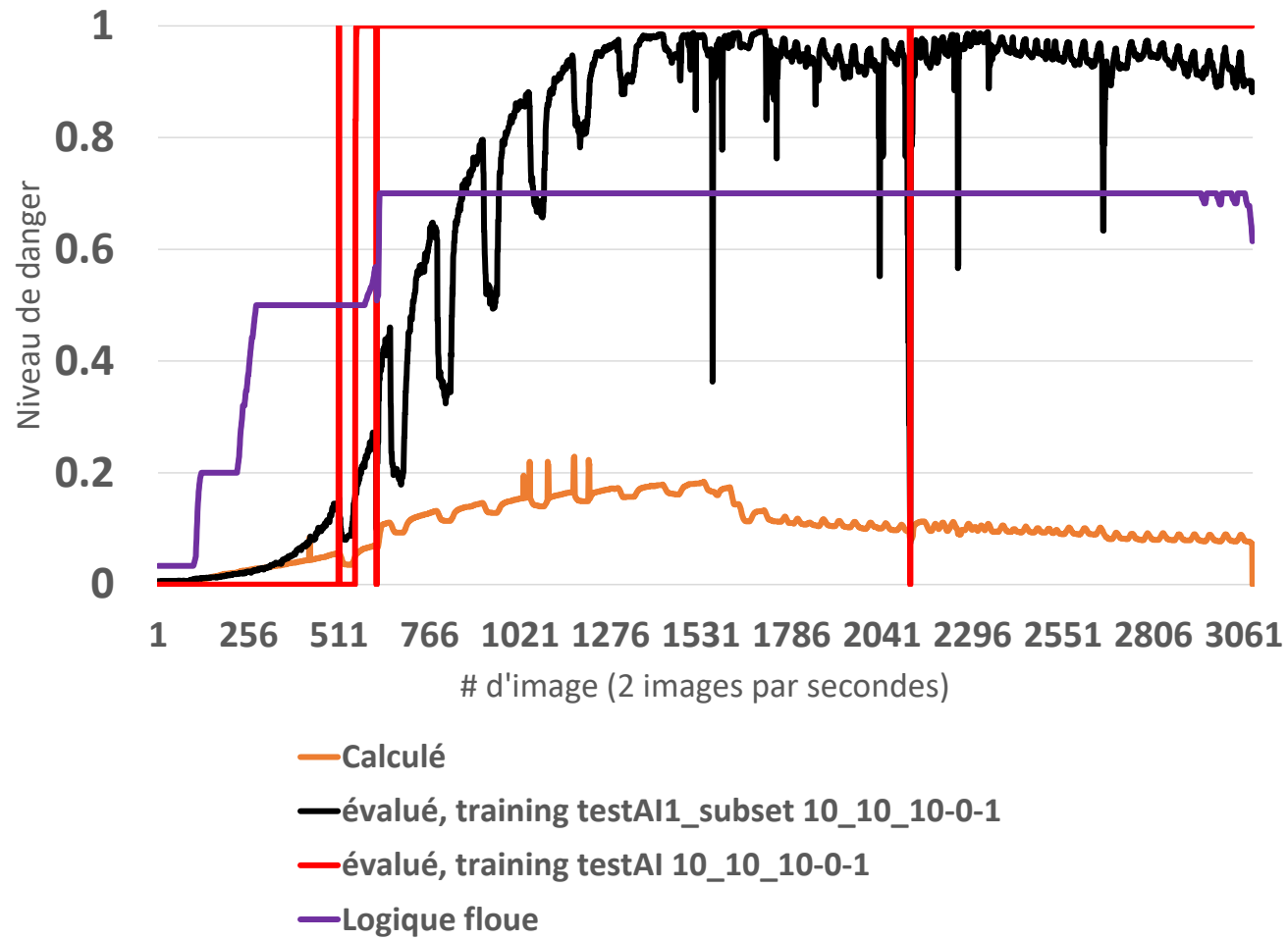


Figure 6.10 Résultat de l'utilisation des deux approches sur « 2016-01-03_18-46-38 »

6.4 Tableaux synthèses

Le tableau 6.1 offre une comparaison entre l'écart de prédiction du système à l'aide de logique floue contre une évaluation manuelle ou un calcul utilisant les mêmes paramètres que la logique floue.

On remarque que l'écart moyen (tel que décrit à l'équation 6.3) est assez important. Une partie de l'explication vient du design des règles de logique floue utilisées. En effet, celles-ci ont été conçues en visant à départager le risque lié à la température d'un risque de brûlure.

$$\frac{\sum |EvaluationManuelle - ReseauNeurones|}{nombreImages} \quad (6.3)$$

Pour ce faire, le niveau de danger lié à la température maximale plafonne à 0,7. Ce n'est qu'en cas d'occlusion que le niveau de danger passe au-dessus.

Bref, la conception des règles rend la prédiction de cette approche plus granulaire (contrairement au réseau de neurones qui tend vers le binaire). On peut donc y voir un avantage : la rétroaction à l'utilisateur sera mieux gradée.

Au niveau de l'écart avec la version calculée, comme il s'agit d'une part d'un calcul (toutes les valeurs de sorties sont possibles) et de l'autre de l'utilisation de fonctions d'appartenance (n'offre en sortie qu'un nombre limité de valeurs), la comparaison est également moins facile.

On remarque tout de même un écart plus petit pour l'utilisation sur l'enregistrement **TestAI1**. Il est donc important de mentionner que le design des règles a été fait à l'aide de ce scénario. Il est donc normal que la performance soit meilleure pour cet enregistrement.

Ensuite, les tableaux suivants (6.2 et 6.1) offrent une vue globale et permettent une comparaison de la performance des différentes approches.

Au tableau 6.2 on peut voir l'écart moyen de prédiction du système sur différents enregistrements en relation avec la source d'entraînement. Rappelons que les prédictions du système varient entre 0 et 1 inclusivement.

On remarque que les écarts sont généralement inférieurs à 10% et que l'utilisation de 3 images consécutives avec un réseau entraîné sur **TestAI** semble offrir un rendement nettement meilleur sur les données de **TestAI1**.

Cependant, on ne peut conclure à une performance nettement supérieure : le réseau entraîné avec une image plutôt que trois ne fait que détecter l'augmentation du niveau de

rappel du nombre d'images	Enregistrement	écart moyen entre évalué et	
		évaluation manuelle (binaire)	calculé
1385	TestAI	55%	48%
140	TestAI1	29%	22%
129	TestAI2	53%	43%
163	Cuisine2	30%	
3085	2016-01-03_18-46-38		53%
	Moyenne	42%	41%

Tableau 6.1 Écart moyen entre les valeurs évaluées manuellement, calculées et l'utilisation de logique floue

danger 2 images en retard. Le reste de l'évaluation sur **TestAI1** (enregistrement très simple avec peu de variations du niveau de danger) reste très similaire.

Comme l'utilisation d'un tel réseau (3 images à la fois) est beaucoup plus demandant (3x 65 données en entrées plutôt que 65) et que les résultats ne sont pas significativement meilleurs, cette approche a été délaissée.

Aussi, on remarque qu'il n'y a pas eu d'évaluation sur **TestAI_subset** autrement que pour valider l'apprentissage sur ces données. Il s'agit d'un choix délibéré puisqu'il s'agit d'un sous-ensemble de **TestAI1** qui lui est évalué).

rappel du nombre d'images	Entrainement	TestAI (1 image)	TestAI (3 images)	TestAI1_subset
	Évaluation			
1385	TestAI	0.7247836%	0.8667108%	7.9408733%
140	TestAI1	1.4285714%	0.0000003%	1.2285915%
18	TestAI1_subset			0.2167256%
129	TestAI2	9.2745334%		19.5397640%
163	Cuisine2	6.7484663%		15.1652694%
	Moyenne	4.5440887%	0.4333555%	8.8182448%

Tableau 6.2 Écart moyen entre les valeurs évaluées manuellement et l'utilisation d'un réseau de neurones

CHAPITRE 7

CONCLUSION

Ce projet aura donc permis de répondre positivement à la question évoquée à la section 3.2 : *Est-ce qu'un patron de température peut permettre d'identifier des situations dangereuses lors de l'utilisation de la surface de cuisson d'une cuisinière électrique ?*

En effet, autant l'approche utilisant la logique floue que celle utilisant un réseau de neurones ont permis de détecter les situations où l'utilisateur est à proximité d'une source de chaleur, donc de détecter les risques de brûlures.

Les deux approches ont également permis de détecter une température élevée, ce qui permet de détecter un risque potentiel d'incendie si un élément inflammable est mis à proximité.

Mais qu'en est-il des autres questions qui ont guidé les expérimentations ? Voici donc un retour sur chacune de ces questions ainsi que les réponses, partielles ou à venir.

- *Quelles sont les limites de température acceptables en toute sécurité ?*

Cette question n'a pas trouvé de réponses directes. On peut considérer qu'une température est acceptable si elle ne crée pas de risques d'incendie ou de brûlure, mais celle-ci sera irréaliste dans un contexte de cuisson.

Peut-être est-ce possible de considérer toute température atteinte lors d'une cuisson acceptable ? Une fois de plus la limite devient difficile à déterminer et ne tient pas compte du danger pour l'utilisateur.

La réponse à cette question pourrait par contre guider un éventuel raffinement des règles utilisées par l'approche utilisant la logique floue.

- *Ces limites peuvent-elles être caractérisées dans certains patrons (spatio-temporels) ?*

Puisque les limites ne sont pas clairement définies, il est difficile de répondre à cette question à l'heure actuelle.

Par contre, on peut affirmer que de tels patrons existent puisque nous en avons utilisé dans les deux approches de traitement.

En effet, la variation des minimums (« blobs » et de l'image complète) en sens inverse pour la logique floue en est un exemple.

Pour l'approche qui utilise un réseau de neurones, la réponse est moins claire puisque « l'intelligence » est « cachée » dans la configuration du réseau et dans les poids des synapses. Cependant, après avoir tenté d'utiliser l'aspect temporel en proposant en entrée trois images à la fois, il semble que ce n'est cet aspect qui permet la détection de dangers avec les configurations des réseaux utilisés.

- *Ces températures et ces patrons sont-ils détectables avec les capteurs ciblés ?*

Oui, la caméra thermique basse résolution MLX90620 de Melexis a en effet permis de fournir au traitement l'information nécessaire.

- *Est-il utile de superviser l'espace environnant immédiat de la surface de cuisson (comptoir par exemple) pour détecter ces patrons ?*

Une réponse partielle à cette question émerge des travaux.

En effet, si l'utilisateur est à proximité de la cuisinière, il s'expose à un risque potentiel de contact avec une source de chaleur. Le risque de brûlure augmente, mais il est difficile de cuisiner sans s'exposer à un minimum de risque.

Au contraire, la présence d'un usager à proximité peut laisser présager que le risque d'incendie est minimisé (une réaction de sa part est possible et la surface de cuisson est probablement surveillée).

Il reste cependant à déterminer la dimension de la zone utile à superviser pour mieux guider le système dans son évaluation de la situation.

- *Est-il possible de détecter la présence d'une main (ou de toutes autres parties du corps) au-dessus (ou à proximité) de la surface de cuisson et de zones de température élevée ?*

Malheureusement il semble difficile, voire impossible, avec le capteur utilisé de déterminer si une main est présente à proximité de la surface de cuisson.

On ne peut que détecter une occlusion, donc que « quelque chose » « passe » entre le capteur et la source de chaleur. On peut alors déduire que l'utilisateur est à proximité, mais pas de son niveau de protection.

Par exemple, pour certaines expérimentations comportant de l'eau à ébullition, donc de la vapeur, le passage d'une main au-dessus de la source de chaleur a été remplacé par le passage d'une main protégée par une mitaine de four. Le système n'a malheureusement pas été en mesure de faire la différence et indique donc à tort le même niveau de danger.

La difficulté réside dans la faible résolution de l'image thermique : une main peut affecter la température de plusieurs pixels, mais la source de chaleur peut également affecter ces derniers (nous n'obtenons donc pas une température proche de celle du corps humain).

7.1 Limites et perspectives

La preuve de concept étant réalisée, il convient de mentionner que nous n'avons pas un prototype fonctionnel prêt à l'utilisation pour assurer la sécurité d'une personne avec déficience cognitive.

Pour y arriver, un raffinement des règles pour la logique floue, si cette approche est retenue, est nécessaire.

Si au contraire on utilise un réseau de neurones, d'autres données d'entraînement devront être produites ou une méthode pour entraîner le système directement avec l'utilisateur devra être prévue.

Finalement, la possibilité évoquée à la section 2.2.3 d'intégrer le prototype développé dans le cadre de cette maîtrise dans un système plus large reste une solution intéressante. Par exemple, le système multi agent proposé par [Kuijpers *et al.*, 2017] pourrait utiliser en tant qu'agent le système développé, ce qui faciliterait l'accès à de l'information complémentaire sur la position de l'utilisateur et sa condition.

ANNEXE A

Exemple de fichier de configuration

Les lignes suivantes donnent un exemple de fichier de configuration utilisé par le moteur d'intelligence artificielle (IA) du projet.

[graphs]

```
width = 10 # per graph
height = 3 # per graph
nbCol = 1 # graph order will be used to place graph from left to
          # right, top to bottom
legend = True
legendSize = small # possible values: xx-small, x-small, small,
                   # medium, large, x-large, xx-large
legendLocation = "upper center" # possible values : upper right,
                                # upper left, lower left, lower right,
                                # right, center left, center right,
                                # lower center, upper center, center
```

[inputs]

[[max]]

```
range = 0, 601, 1
label = 'Max Temp'
graphOrder = 0
```

[[[MF]]]

[[[low]]]

```
type = trimf
params = 0, 0, 100
color = 'g'
label = 'Low'
```

[[[med]]]

```
type = trimf
params = 0, 100, 200
color = 'b'
label = 'Medium'
```

[[[hi]]]

```
type = trapmf
params = 100, 200, 600, 600
color = 'r'
label = 'Hi'
```

```
[outputs]
```

```
[[danger]]
    range = 0, 10, 1
    label = 'Danger Level'
    graphOrder = 1
```

```
[[[MF]]]
    [[[low]]]
        type = trimf
        params = 0, 0, 5
        color = 'g'
        label = 'Low'
```

```
    [[[med]]]
        type = trimf
        params = 0, 5, 10
        color = 'b'
        label = 'Medium'
```

```
    [[[hi]]]
        type = trimf
        params = 5, 10, 10
        color = 'r'
        label = 'High'
```

```
[rules]
```

```
[[1]]
```

```
[[[if]]]
```

```
    max = low
```

```
[[[then]]]
```

```
    danger = low
```

```
[[2]]
```

```
[[[if]]]
```

```
    max = med
```

```
[[[then]]]
```

```
    danger = med
```

```

[[3]]

[[[if]]]

    max = hi

[[[then]]]

    danger = hi

[SECUIUS]

variableBlobThreshold = False # True or False
blobThreshold = 75 # Celcius for blob detection, used only
                        # if variableBlobThreshold is False
filterDepth = 10 # nb of frames to use to filter the data
filterDepthBlob = 3 # nb of frames to use to filter blob data

```

Vous constaterez qu'il existe 5 sections principales :

graphs Paramètres en lien avec l'affichage de graphiques (intimement lié avec le visualisateur).

inputs Paramètres des images thermiques à considérer comme entrées pour l'IA.

outputs Ce que l'on souhaite obtenir comme sortie, donc résultat, du moteur d'IA.

rules Les règles qui permettent d'obtenir les sorties à partir des entrées.

SECUIUS Quelques suppléments permettant de guider l'extraction des paramètres à partir de l'image thermique.

Dans la majorité de ces sections, des sous-sections s'imbriquent. Par exemple, pour les entrées et les sorties, on voit apparaître des « `[[max]]` » et « `[[danger]]` ». Il s'agit, pour l'entrée, d'utiliser un nom de paramètre valide (voir la liste des paramètres à la section [B](#)) et pour la sortie de créer un nom qui sera ensuite réutilisé dans les règles.

L'imbrication ne s'arrête cependant pas à 2 niveaux. Vous remarquerez qu'à chaque niveau, il est possible de trouver des paires clé/valeur ainsi que d'annoncer une sous-section.

Ainsi, pour chaque entrée ou sortie, on précise la plage (range), un nom (pour les graphiques) et l'ordre d'affichage (pour les graphiques). On définit ensuite les paramètres de la fonction d'appartenance (MF).

Dans le cas des règles, il suffit d'utiliser un nom unique à chaque règle et ensuite préciser une condition (if) et conséquence (then). Pour ce faire, il est possible d'utiliser dans la condition différents opérateurs logiques (et(and) / ou (or) / non (not)) pour créer des conditions plus complexes.

Comme une règle doit lier les entrées aux sorties, une condition doit utiliser au moins une entrée (avec la fonction d'appartenance désirée) et au moins une conséquence (une sortie et la fonction d'appartenance à activer).

ANNEXE B

Description des paramètres

average Moyenne des 64 valeurs de température mesurées par le capteur pour une image thermique.

blobsAverage Pour chaque zone (« blob »), une moyenne de la température des pixels participants est calculée.

blobsGlobalAverage Nombre unique représentant la moyenne de température dans les « blobs ».

blobsGlobalMax Nombre unique représentant le maximum de température dans les « blobs ».

blobsGlobalMin Nombre unique représentant le minimum de température dans les « blobs ».

blobsMax Pour chaque zone (« blob »), le maximum de la température des pixels participants est calculé.

blobsMin Pour chaque zone (« blob »), le minimum de la température des pixels participants est calculé.

delta_average Variation temporelle de la moyenne (« average »).

delta_blobsGlobalAverage Variation temporelle de la moyenne de température des « blobs » (« blobsGlobalAverage »).

delta_blobsGlobalMax Variation temporelle du maximum de température des « blobs » (« blobsGlobalMax »).

delta_blobsGlobalMin Variation temporelle du minimum de température des « blobs » (« blobsGlobalMin »).

delta_max Variation temporelle du maximum (« max »).

delta_min Variation temporelle du minimum (« min »).

delta_overThresholdBlobsNb Variation temporelle du nombre de « blobs » (« overThresholdBlobsNb »).

delta_overThresholdNb Variation temporelle du nombre de pixels au-dessus du seuil pour faire partie d'un « blob » (« overThresholdNb »).

filtered_average Moyenne (« average ») filtrée sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_blobsGlobalAverage Moyenne globale des « blobs » (« blobsGlobalAverage ») filtrée sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_blobsGlobalMax Maximum global des « blobs » (« blobsGlobalMax ») filtré sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_blobsGlobalMin Minimum global des « blobs » (« blobsGlobalMin ») filtré sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_max Maximum (« max ») filtré sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_min Minimum (« min ») filtré sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_overThresholdBlobsNb Nombre de « blobs » (« overThresholdBlobsNb ») filtré sur un nombre d'images thermique défini dans le fichier de configuration.

filtered_overThresholdNb Nombre de pixels dont la température excède le seuil défini dans le fichier de configuration (« overThresholdNb ») filtrée sur un nombre d'images thermique, également défini dans le fichier de configuration.

max Maximum des 64 valeurs de température mesurées par le capteur pour une image thermique.

min Minimum des 64 valeurs de température mesurées par le capteur pour une image thermique.

overThresholdBlobs Liste des « blobs ». Chaque « blob » y est représenté par la liste de ses pixels.

overThresholdBlobsNb Nombre de « blobs » détectés.





















overThresholdIndexes Index des pixels dont la température excède le seuil (déterminé dans le fichier de configuration) et qui fait donc parti d'un « blob ».

overThresholdNb Nombre de pixels dépassant le seuil pour faire partie d'un « blob ».

ANNEXE C

Configuration des GPIO du Raspberry Pi 2, modèle B

¹Figure disponible à l'adresse suivante : <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Raspberry Pi 2 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART) 15
	Ground	9		10	GPIO 16 RxD (UART) 16
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0 1
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4 4
	3.3 VDC Power	17		18	GPIO 5 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI) 10
	Ground	25		26	GPIO 11 CE1 (SPI) 11
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM) 31
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0 26
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN 28
	Ground	39		40	GPIO 29 PCM_DOUT 29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Figure C.1 Disposition des broches (« GPIO pinout »), Raspberry Pi 2, modèle B¹

LISTE DES RÉFÉRENCES

- (2017). <https://www.json.org/> (page consultée le 2017-11-11). 61
- Ahrens, M. (2008). Home smoke alarms : The data as context for decision. 2, 9
- Ahrens, M. (2012). *Home Fires Involving Cooking Equipment* (Rapport technique). National Fire Protection Association, 217 p. 1
- Ahrens, M. (2013). *Home Structure Fires* (Rapport technique). National Fire Protection Association, 146 p. 1
- Alzheimer's Association (2009). *Alzheimer's & dementia : the journal of the Alzheimer's Association*, chapitre 2009 Alzheimer's disease facts and figures, *volume 5*. Elsevier, Inc., p. 234–270. 5
- Bharucha, A. J., Anand, V., Forlizzi, J., Dew, M. A., Reynolds, 3rd, C. F., Stevens, S. et Wactlar, H. (2009). Intelligent assistive technology applications to dementia care : current capabilities, limitations, and future challenges. *The American journal of geriatric psychiatry : official journal of the American Association for Geriatric Psychiatry*, volume 17, numéro 2, p. 88–104. 8
- Bonner, Brian, C. (2007). *Strategies to reduce home kitchen fires*. National Fire Academy, 40 p. 11, 12
- Bosch Sicherheitssysteme GmbH (2014). <http://www.boschsecurity.com/> (page consultée le 2014-03-10). 10
- Bosch Sicherheitssysteme GmbH (2017a). http://us.boschsecurity.com/en/products/firearmsystems/en54_1/automaticfiredetectors_1/pointtypedetectors_1/fcp320fch320conventionala_1/fcp320fch320conventionala_1_24711 (page consultée le 2017-11-18). 10
- Bosch Sicherheitssysteme GmbH (2017b). https://us.boschsecurity.com/en/products/firearmsystems/uslulcfm/conventionalproducts/conventionaldetectorsbase/fcp500conventionalflushmo/fcp500conventionalflushmo_25116 (page consultée le 2017-11-18). 10
- Bouchard, D. (2013). *Documentation sur l'utilisation Interface Domus UdeS*. Sermax Automation Inc. 5
- Chen, S., Bao, H., Zeng, X. et Yang, Y. (2003). A fire detecting method based on multi-sensor data fusion. Dans *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. volume 4. IEEE, p. 3775–3780. 10
- Crowley, M. (2004). Stopping a fire before it starts. *National Fisherman*, volume 84, numéro 12, p. 65. 11

- de Champs, T. (2012). *Approche a base de verification formelle de modele pour une utilisation securitaire de la cuisiniere d'un habitat intelligent*. Mémoire de maîtrise, Université de Sherbrooke, Sherbrooke, Québec, Canada, 148 p. 8, 11, 12
- Dreiseitl, S., Osl, M., Scheibböck, C. et Binder, M. (2010). Outlier detection with one-class svms : An application to melanoma prognosis. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, volume 2010, p. 172. 16
- Ehrlich, A. R., Bak, R. Y., Wald-Cagan, P. et Greenberg, D. F. (2008). Risk factors for fires and burns in homebound, urban elderly. *Journal of burn care & research : official publication of the American Burn Association*, volume 29, numéro 6, p. 985. 1
- Ehrlich, A. R., Kathpalia, S., Boyarsky, Y., Schechter, A. et Bijur, P. (2005). Elderly patients discharged home from the emergency department with minor burns. *Burns : journal of the International Society for Burn Injuries*, volume 31, numéro 6, p. 717–720. 1
- Farmer, A. R. (2012). *Stove Top Fire Suppression*. National Fire Academy, 78 p. 11, 12
- FEMA (2004). *Kitchen Fires* (Rapport technique). U.S. Fire Administration/National Fire Data Center, 6 p. 1
- Fischer, D. (2004). *Artificial intelligence techniques applied to fault detection systems*. Mémoire de doctorat, McMaster University, 1280 Main St W, Hamilton, ON L8S 4L8, 195 p. 16, 17, 36
- Geoff West, C. N. et Greenhill, S. (2005). Using a camera to implement virtual sensors in a smart house. Dans Sylvain Giroux, H. P., *From Smart Homes to Smart Care*. IOS Press, p. 83–90. 5
- Gottuk, D. T., Peatross, M. J., Roby, R. J. et Beyler, C. L. (2002). Advanced fire detection using multi-signature alarm algorithms. *Fire Safety Journal*, volume 37, numéro 4, p. 381 – 394. 10
- Hall, J. R. (2006). *Home cooking fire patterns and trends*. National Fire Protection Association, 126 p. 1
- Henriques, F. C. et Moritz, A. R. (1947). Studies of thermal injury : I. the conduction of heat to and through skin and the temperatures attained therein. a theoretical and an experimental investigation. *The American journal of pathology*, volume 23, numéro 4, p. 530–549. 14
- HomeSense Enterprises (2006). <http://www.homesensers.com/> (page consultée le 2014-03-04). 11
- Istre, G. R. et Mallonee, S. (2000). Commentary : Smoke alarms and prevention of house-fire-related deaths and injuries. *Western Journal of Medicine*, volume 173, numéro 2, p. 92. 2

- Kuijpers, N., Philippe, J.-L. et Giroux, S. (2017). *Système autonome de sécurité lors de la préparation d'un repas pour des personnes cognitivement déficientes dans un habitat intelligent pour la santé*. Université de Sherbrooke. 8, 9, 91
- Laboratoire DOMUS (2010). <http://domus.usherbrooke.ca/> (page consultée le 2014-02-25). 5
- Larochelle, H. (2014). http://www.dmi.usherb.ca/~larocheh/cours/ift725_A2014/contenu.html. 36, 38
- Mallonee, S., Istre, G. R., Rosenberg, M., Reddish-Douglas, M., Jordan, F., Silverstein, P. et Tunell, W. (1996). Surveillance and prevention of residential-fire injuries. *The New England journal of medicine*, volume 335, numéro 1, p. 27–31. 8
- Melexis (2014). <http://www.melexis.com/Infrared-Thermometer-Sensors/Infrared-Thermometer-Sensors/MLX90620-776.aspx> (page consultée le 2014-03-10). 19
- Moritz, A. R. (1947). Studies of thermal injury : Iii. the pathology and pathogenesis of cutaneous burns. an experimental study. *The American journal of pathology*, volume 23, numéro 6, p. 915–941. 14
- Moritz, A. R. et Henriques, F. C. (1947). Studies of thermal injury : Ii. the relative importance of time and surface temperature in the causation of cutaneous burns. *The American journal of pathology*, volume 23, numéro 5, p. 695–720. 14
- Negnevitsky, M. (2005). *Artificial intelligence : a guide to intelligent systems*. Addison-Wesley, New York, 435 p. 15, 28, 36, 39
- Nest Labs (2014). <https://nest.com/ca/fr/smoke-co-alarm/life-with-nest-protect/> (page consultée le 2014-03-10). 2, 10
- Paszczuk, H. (2009). *Reducing the risk from cooking fires for occupants of multiple family dwellings in the City of New Britain*. National Fire Academy, 85 p. 11, 12
- Pfister, G. (1997). Multisensor/multicriteria fire detection : A new trend rapidly becomes state of the art. *Fire technology*, volume 33, numéro 2, p. 115. 10, 16
- Pioneering Technology Corp. (2011). <http://www.pioneeringtech.com/safe-t-element> (page consultée le 2014-03-12). 11
- Protection E.M. Inc. (2012). <http://www.stoveguardintl.com/> (page consultée le 2014-03-12). 11
- Protection E.M. Inc. (2014). <http://www.safecook.ca/> (page consultée le 2014-03-12). 11
- Reed, R. (2014). Thermocouple junctions are not voltage sources. 25
- Rose-Pehrsson, S. L., Shaffer, R. E., Hart, S. J., Williams, F. W., Gottuk, D. T., Strehlen, B. D. et Hill, S. A. (2000). Multi-criteria fire detection systems using a probabilistic

- neural network. *Sensors and Actuators B : Chemical*, volume 69, numéro 3, p. 325 – 335. 10
- Russell, S. J. S. J., Norvig, P. et Davis, E. (2010). *Artificial intelligence : a modern approach*. Prentice Hall, Upper Saddle River, N.J, 1142 p. 15, 36, 38, 44
- StoveTop FireStop (2014). <http://www.stovetopfirestop.com/> (page consultée le 2014-03-04). 12
- Tessier, T. (2012). *Prévention des dangers liés à l'utilisation d'une cuisinière : une validation grâce à un prototype*. Mémoire de maîtrise, Université de Sherbrooke, Sherbrooke, Québec, Canada, 114 p. 6, 8, 11, 12, 13, 14, 15, 16, 17, 20
- The Raspberry Pi Foundation (2017a). <https://www.raspberrypi.org/documentation/hardware/camera/README.md> (page consultée le 2017-11-18). 49
- The Raspberry Pi Foundation (2017b). <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/> (page consultée le 2017-08-04). 52
- Wartzek, T., Muhlsteff, J. et Imhoff, M. (2011). Temperature measurement. *Biomedizinische Technik / Biomedical Engineering*, volume 56, numéro 5, p. 241. 14, 29
- Welch, S. (2015-2016). <https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRaiPoU>. 36, 39, 44
- Wijayasinghe, M. S. et Makey, T. B. (1997). Cooking oil : A home fire hazard in alberta, canada. *Fire technology*, volume 33, numéro 2, p. 140. 13
- Yin, J., Yang, Q. et Pan, J. (2008). Sensor-based abnormal human-activity detection. *Knowledge and Data Engineering, IEEE Transactions on*, volume 20, numéro 8, p. 1082–1090. 16

